MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

SECURITY AND CRYPTOGRAPHY IN BSD

INSIDE

MAHESHAOPENBSD - OPENBSD IN A SERPENT WORLD

SECURITY POLICY DEVELOPMENT IN TRUSTEDBSD MAC FRAMEWORK

USING NGINX AND NAXSI FOR SECURITY AND HIGH AVAILABILITY

FREEBSD PROGRAMMING PRIMER – PART 6

ELLIPTIC CURVE CRYPTOGRAPHY AND BASIC APPLICATIONS IN BSD OS

VOL.7 NO.7
ISSUE 07/2013(48)
1898-9144

Dear Readers,

This issue of BSD Magazine is dedicated to security concepts including Elliptic Curve Cryptography, Security Policy Development in TrustedBSD MAC Framework and Naxsi, the Nginx Web Application Firewall. Moreover, you will find articles about MaheshaOpenBSD and CSS programming.

We start with the What's New section, where Juraj Sipos shares with you his thoughts on a turnkey server made with OpenBSD.

Then, Mohamed Farag describes the overall process of developing and applying different security policies within the FreeBSD kernel under the TrustedBSD MAC Security Framework.

Next, Alexandro Silva talks a bit about Nginx, which is a lightweight and powerful HTTP proxy, and Naxsi, which is a Nginx Web Application Firewall.

This month's Admin section covers designing a basic menu navigation system and styling it with CSS.

Finally, in the Cryptography section, José B. Alós discusses how to protect against intruders using Elliptic Curve Cryptography.

We hope you enjoy this issue and find many interesting articles inside!

Kamil Sobieraj
Editor of BSD Magazine
& BSD Team

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa.dudzic@software.com.pl

Supportive Editor:
Kamil Sobieraj
kamil.sobieraj@software.com.pl

Contributing:
Juraj Sipos, Mohamed Farag, Alexandro Silva, Rob Somerville, José B. Alós

Top Betatesters & Proofreaders:
Ben Milman, Annie Zhang, Luca Ferrari, Imad Soltani, Eric Geissinger, Barry Grumbine

Special Thanks:
Denise Ebery
Luca Ferrari

Art Director:
Ireneusz Pogroszewski

DTP:
Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Senior Consultant/Publisher:
Paweł Marciniak
pawel@software.com.pl

CEO:
Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director:
Andrzej Kuca
andrzej.kuca@software.com.pl

Advertising Sales:
Kamil Sobieraj
kamil.sobieraj@software.com.pl

Publisher:
Software Media Sp. z o.o. SK
ul. Bokserska 1, 02-682 Warszawa
Poland
worldwide publishing
tel: 1 917 338 36 31
www.bsdmag.org

Software Media Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

Mathematical formulas created by Design Science MathType™.

Let's Talk

What's New

Admin

Cryptography

An Eye For An Eye Soon Leads To Blindness

Former National Security Agency contractor, Edward Snowden, has created worldwide controversy by leaking United States secrets to British and American newspapers. While Snowden seeks permanent asylum, the US bans access to the UK Guardian newspaper for US troops. This follows in the footsteps of the Stuxnet attack on Iranian nuclear infrastructure. Are we at war on the web?

With human nature being what it is, no technological innovation can remain unsullied from the fingerprints of those with less than pure motives. Pornographers hijacked the fledgling movie industry in the early 20th century and continued with the next innovation, video. The same ethos applies to conmen, thieves, hustlers, fraudsters and many others that inhabit and exploit the power of the Internet. Clearly, there will always be a dark side. This contrasts strongly with the whole idea of the Internet and the World Wide Web being a platform for openness, democracy, free speech, etc.

Then again, there are institutions, schools, corporations, etc. who view the Internet as just an extension to their Local Area Network – joining satellite offices together transparently. While some use the medium to express themselves and communicate, others prefer to watch silently in the background and monitor trends, patterns and metadata. Some attack vulnerable systems for laughs, others for commercial gain. All of this is understood. Where the picture becomes unclear is how much crossover there is between the military and intelligence services (e.g. Defence Advanced Projects Research Agency – DARPA and the National Security Agency – NSA) and the Internet user. Just how much traffic is monitored? Is it content or metadata (e.g. IP address to IP address)? How long is it held for? How much power do the watch-

ers have to examine my traffic, my exchanges in detail? Is this cross-border, across nations?

A lot has changed since the formation of DARPA in 1958. The Cold War is over, and nations, rather than strengthening on the basis of individuality, are coalescing into large trading groups. The world is becoming smaller; consumerism and Western values are penetrating nations who, 50 years ago, would have rejected such principles. With connectivity via satellite, the World Wide Web literally encompasses the whole world, be it urban, suburban or inhospitable jungle or desert. The Internet is not limited to PC's or servers, mobile devices (even domestic appliances and cars) are online. A new generation has embraced social media with gusto, impervious to the fact that in 30 years time, the inane, immature and sarcastic will be available to all as well as the more illustrative or insightful. Ironically, while technology seems to have moved forward, the terrorist attacks of 2001 have spawned a culture of security and watchfulness that – it could be argued – is more restrictive than during the 50's or 60's.

Throughout history, the powers that be (TPTB) have always used some method to monitor communications. The meta-data (who called who) for telephone calls in the UK at least, has to be stored for a number of years. Paper mail can be intercepted with a warrant, and the tales of bugs and microphones turning up in embassies throughout the world are legion. Everybody knows that Internet traffic can be easily monitored (ask any webmaster) – the ethical question is "How much can we trust those that watch?" It is all very well to say that if you have nothing to hide, you have nothing to fear, but in intelligence circles, there is no smoke without fire, and guilt by association – while not sufficient evidence to convict in a court of law – could assign an individual's name to some secure database somewhere, with no right of appeal or correction.

The generic term "In the interests of National Security" covers a multitude of sins, and looking to other countries, quite frequently servers in the West will be attacked either for political, espionage or commercial reasons. What might be good for one nation will be bad for another, so the battle rages on, the weapons get bigger, the deployments at the battle-front more numerous. Legal pre-emptive hacking strikes have been written into Singapore law (The Computer Misuse and Cybersecurity Act). We have now entered a vicious cycle of a technological arms race via the Internet and Local Area Networks; penetrating the Iran nuclear facility with the Stuxnet worm – an act considered science fiction during the Cold War– was three years ago. While the delivery vector was probably a USB enabled device, the level of sophistication suggests military, intelligence or criminal origins rather than your run of the mill hacker.

Besides controlling who has access to content, there is the separate issue of what content is acceptable. There used to be a 9pm watershed for all broadcast media – anything that might upset or cause offence could not be televised before then. With the introduction of the VCR and PVR, this control is now obsolete. As for the Internet, it's 24/7 whatever you want. It looks like the argument my father used is appropriate here: "There is always the Off button". There are already calls for Internet censorship, and like the history of the book, cinema and printing press, we are approaching the time when some web pages will be stacked high, condemned and then publicly burnt. The Internet has suddenly become very political.

The Internet and the World Wide Web as a force for good cannot be denied, yet as a technology, it is still relatively immature. Colour television was available in the 1970's, but mass adoption of the browser did not happen until 25 years later. What we are experiencing are the growing pains – as the infrastructure becomes more essential to corporations, etc., more regulation will be put into place and in 20 years' time, the Internet will be a very different place.

As with Wikileaks, the Snowden incident will no doubt take years to come to a legal and political conclusion. Meanwhile, tit-for-tat actions and attacks will continue to escalate whether they come from individual hackers, organised crime, government agencies or the military. We should focus on strengthening infrastructure rather than getting distracted by politicians.

It has always been war on the Internet. It is just that we have moved on from using words to throwing rocks, sticks and using much larger scale weapons.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

MaheshaOpenBSD

– OpenBSD In A Serpent World

The article focuses on a turnkey server made with OpenBSD. This is available as a USB image that can be immediately deployed in the SOHO (Small Office – Home Office) environment. MaheshaOpenBSD Server is a flavor of the MaheshaBSD Server, originally based on FreeBSD, that runs on OpenBSD 5.2 and was released on April 5, 2013.

What you will learn…

- how to set up a quick and secure server for SOHO purposes
- how to use OpenBSD quickly anywhere in the world

What you should know…

- basic use of OpenBSD
- basic shell commands

This article will teach anybody how to deploy a secure server using OpenBSD on a USB flash drive and how to set up the secure server in a few seconds without installing OpenBSD onto your hard drive.

Welcome Into The World Of Serpents

The Moon is the symbol of Manasa, the Hindu Tantric Serpent Goddess.

An article entitled "MaheshaBSD Server: MySQL and WordPress in FreeBSD" was published in issue 44 of BSD Magazine. Now, the project MaheshaBSD has been extended to all major flavors of BSD distributions including NetBSD, OpenBSD and DragonFlyBSD. Not many BSD systems (like NetBSD) were previously available with preinstalled software as USB images, thus this project has an educational purpose too. At present, the MaheshaBSD Project has two versions:

- General-purpose and free MaheshaBSD USB/CD distribution (FreeBSD, NetBSD, DragonFlyBSD).
- Server version of MaheshaBSD (FreeBSD and OpenBSD), which is free for personal use.

At the time of writing, the MaheshaBSD Project consists of:

- MaheshaBSD – a general purpose LiveCD distribution based on FreeBSD 9.0-RELEASE.
- MaheshaBSD Server – the same thing as MaheshaBSD, but this is the server version with additional software common to server deployments like WordPress, MySQL, phpMyAdmin, etc. installed. This edition is only available as a USB image.
- MaheshaNetBSD – a general purpose NetBSD USB distribution.
- MaheshaDragonFlyBSD – a general purpose DragonFlyBSD USB distribution.
- MaheshaOpenBSD Server – the subject of this article, an OpenBSD based Mahesha server distribution.

Releases of these BSD systems are downloadable (see the links). The MaheshaBSD Project is copyrighted software.

Introduction

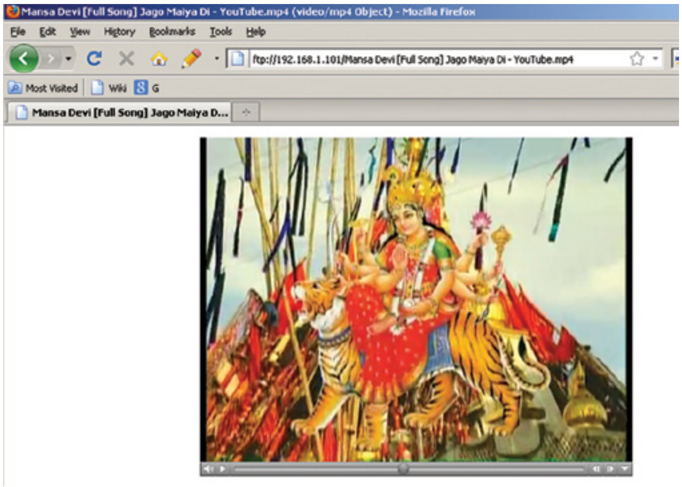
A pre-configured turnkey WordPress / FTP Server on a bootable USB image is not easy to find on the Internet, even on Linux specific sites. MaheshaOpenBSD runs on a writable USB memory stick that can be immediately deployed just by writing the image onto your USB flash drive (At least a 4 GB memory stick is required). The project's goals are education, simplicity, BSD advocacy, and a focus on poorer countries. The MaheshaBSD slogan is KEEP BSD SIMPLY STUPID AND TANTRIC. Most (if not all) online FTP servers have restrictions and bypassing them requires payment. With MaheshaBSD Server, anybody can use any hard disk and set up their own FTP/WordPress server anywhere in the world. MaheshaBSD Server is a social project too as people can easily share their files. Students may put a link on their personal websites to a MaheshaBSD FTP/WordPress server running at their home and share their files (videos, MP3's, etc.) with anybody in the world (Figure 1).

MaheshaOpenBSD Server is the same thing as MaheshaBSD Server. Both MaheshaBSD Server (FreeBSD) and MaheshaOpenBSD Server (OpenBSD) will offer users the same functionality – that is, an almost identical desktop interface with IceWM, a turnkey WordPress and FTP server with MySQL. Both systems can also run from a USB flash drive. You can make an ISO image with MaheshaOpenBSD and use it in conjunction with alternate deployment strategies.

A Quick Overview Of MaheshaOpenBSD Server

MaheshaOpenBSD Server is a server that behaves like any website running on any server that you access on the Internet. You do not need any special knowledge of Unix – you will just log in, copy or remove files, and that's it (Figure 2).

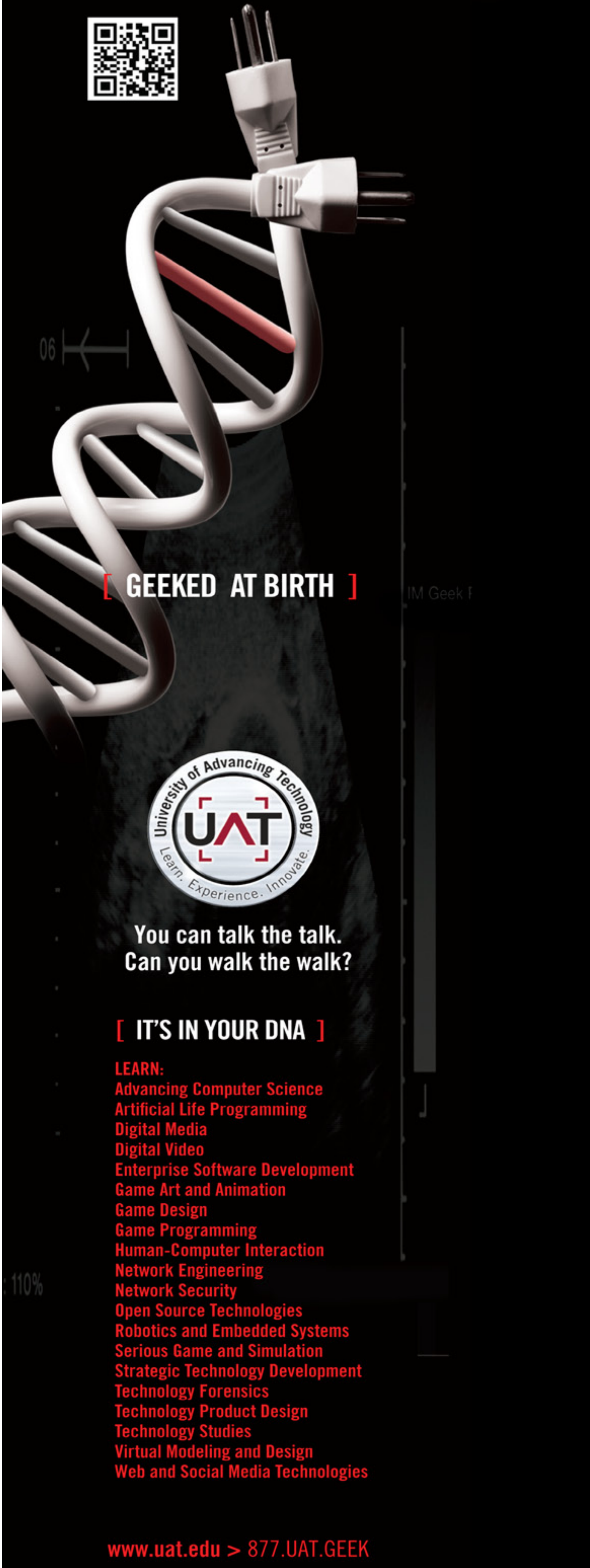
If you have a Facebook account, nothing will be easier than just sharing a link to this SOHO server (usually it is not easy and definitely not cheap to share tons of files on the Internet). All you need to do is enable the IP Forwarding feature in your router. With a spare notebook, you will have a home server with tons of files that you can instantly share with friends as MaheshaOpenBSD Server does not need any special setup. WordPress behaves the same way like in MaheshaBSD Server (FreeBSD). You can keep your colleagues or family updated with your progress. Provided you ran the `dhclient` on your network card in MaheshaOpenBSD, the `ifconfig` command will show you which IP to use. Use this IP in your browser, or alter-


Mansa Devi [Full Song] Jago Maiya Di - YouTube.mp4 (video/mp4 Object) - Mozilla Firefox

ftp://192.168.1.101/Mansa Devi [Full Song] Jago Maiya Di - YouTube.mp4

Mansa Devi [Full Song] Jago Maiya D...

Figure 1. *With MaheshaOpenBSD Server you may watch videos over the network*



natively change the hosts file on the computer from which you will access MaheshaOpenBSD Server:

```
192.168.1.101 manasa
```

The hosts file in Windows resides in C:\Windows\System32\drivers\etc (on almost all versions of Windows).

If you edited your hosts file as mentioned above, Apache server will then run on: *http://manasa*. FTP server will run on: *ftp://manasa*. WordPress will be accessible at: *http://manasa/wordpress* (Figure 3).

To listen to audio files or to play video files over the network, you must have the Flash Plugin installed in your browser on the computer from which you will access MaheshaOpenBSD Server. With files converted to the MP4 format, you can start your own SOHO "YouTube" alternative. To convert files to the MP4 format, just use any free online video conversion tool.

Quick Start

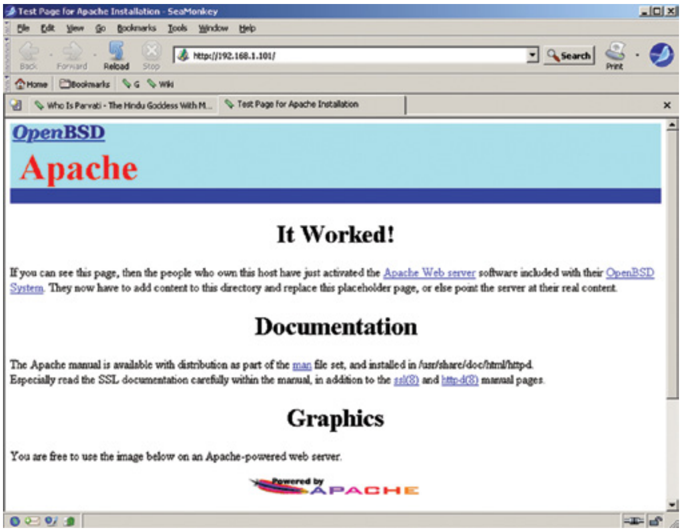
The basic steps detailing how to use this software are described in the README!!!.txt in the RAR file that is available on the Internet at *ftp://2227.x.rootbsd.net/index.html*.

Download the image, unrar it and use the following command (OpenBSD) to write it onto your flash drive (assuming the flash drive has been attached to `/dev/sd0` – check before executing the command):

```
dd if=MaheshaOpenBSD.ima of=/dev/sd0 bs=10240 conv=sync
```

If you use Windows, use a program such as WinImage.

Boot the image: reboot your computer to ensure that the USB boot is enabled.


Test Page for Apache Installation - SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://192.168.1.101/
Search
Print
Home Bookmarks G Wiki
Who Is Parvati - The Hindu Goddess With M...
Test Page for Apache Installation
OpenBSD
Apache
Powered by APACHE

It Worked!

If you can see this page, then the people who own this host have just activated the Apache Web server software included with their OpenBSD System. They now have to add content to this directory and replace this placeholder page, or else point the server at their real content.

Documentation

The Apache manual is available with distribution as part of the man file set, and installed in /usr/share/doc/html/httpd.
Especially read the SSL documentation carefully within the manual, in addition to the ssl(8) and httpd(8) manual pages.

Graphics

You are free to use the image below on an Apache-powered web server.

Figure 2. *Just replace the Apache server's default files in /var/www/htdocs*

Figure 3. *The boss account in Windows accessible via samba*

Log in to MaheshaOpenBSD Server physically as *root* with password *manasa11* (change the root password as soon as possible). Run the `dhclient` command to get a new IP address from your local network. For instance if your network card is `fxp0`, type the following in your shell: `dhclient fxp0.` Edit the hosts file on your host computer to add the IP of your MaheshaOpenBSD Server. This will enable you to access MaheshaOpenBSD Server's services (but this is required only for WordPress). That's all.

How To Use FTP/WWW Server In MaheshaOpenBSD Server?

FTP

To operate the FTP server immediately, log in to MaheshaOpenBSD Server's vsftpd account via SFTP (password: 717manageme – passwords for all accounts are in `/home/guest5/passes.txt`). You may also use a free Windows program like WinSCP (Figure 4).

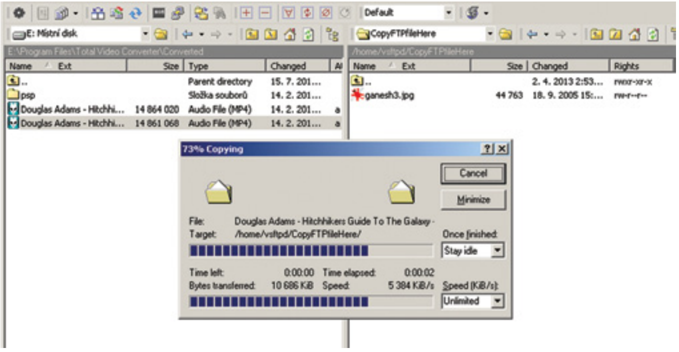
As you see, WinSCP works very well. Copy anything to `CopyFTPfileHere` directory – `/home/vsftpd/CopyFTPfileHere`, then log in via SSH to MaheshaOpenBSD Server's `guest5` account, (password is guest6), fetch your passwords and then log in as "boss" via SSH. `su` to root and copy that file from `CopyFTPfileHere` in /home/vsftpd to `/home/vsftpd/ftp`. This seems a little bit more complicated than in MaheshaBSD Server (FreeBSD), but this is because VSFTPD behaves differently in OpenBSD.

The configuration file for VSFTPD (vsftpd.conf) in MaheshaOpenBSD Server will help beginners learn about this different behavior. Contrary to FreeBSD, OpenBSD stores this file in `/etc`.

WWW/WordPress

WWW

Just copy *.php, *.htm, or anything to `/var/www/htdocs`.



E:\Program Files\Total Video Converter\Converted

Name	Ext	Size	Type	Changed	At
..			Parent directory	15. 7. 201...	
psp			Složka souborů	14. 2. 201...	
Douglas Adams - Hitchhi...		14 864 020	Audio File (MP4)	14. 2. 201...	a
Douglas Adams - Hitchhi...		14 861 068	Audio File (MP4)	14. 2. 201...	a

/home/vsftpd/CopyFTPfileHere

Name	Ext	Size	Changed	Rights
..			2. 4. 2013 2:53...	rwxr-xr-x
ganesh3.jpg		44 763	18. 9. 2005 15:...	rw-r--r--

73% Copying

File: Douglas Adams - Hitchhikers Guide To The Galaxy -
Target: /home/vsftpd/CopyFTPfileHere/
Once finished: Stay idle
Time left: 0:00:00 Time elapsed: 0:00:02
Bytes transferred: 10 686 KiB Speed: 5 384 KiB/s
Speed (KiB/s): Unlimited

Figure 4. *Files to and from MaheshaOpenBSD Server are copied with the SFTP file protocol*

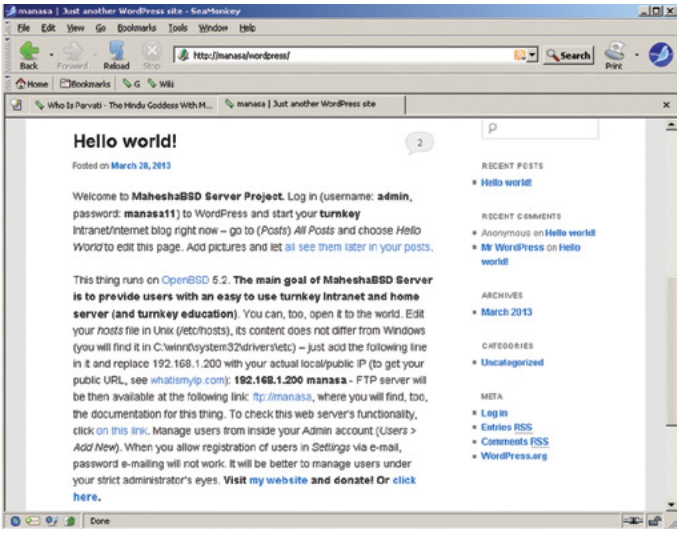
WordPress

As already stated, you must edit the hosts file on your host computer as WordPress needs it in its settings. It would be unpleasant to have to change it every time because of dynamic IP's, etc (Figure 5).

WordPress is quite easy to use. Add pictures, files, or create new links, etc. after you log in. To change the introductory screen, edit Hello world! in your Posts (Figure 6).

Some Notes And Tips

MaheshaOpenBSD Server has espeak (text to speech conversion software) and a number of other useful utilities. New packages can be installed easily (from the Internet) – just type `pkg-get -i package` in your shell. `Pkg-get` is a useful script, as it will also help you search for packages. For example, if you type `pkg-get -s fire` in your shell, you will see all packages available with the prefix "fire". To install a package, just select its number and press Enter. A number of packages are installed in MaheshaOpenBSD Server. You can see them all in `/pkg`. To use X, type `startxaut` (the script will generate `/etc/X11/xorg.conf` and will start your X session automatically). You may also learn Sanskrit. To write documents, I decided to use the Seamonkey's Composer Component – its icon is on the IceWM desktop like in all MaheshaBSD projects. Keyboard layouts are available in the IceWM menu – Start > Utilities.


manasa | Just another WordPress site - SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
Search
Print
Home
Bookmarks
G
Wiki
Who Is Parvati - The Hindu Goddess With M...
manasa | Just another WordPress site
Done

http://manasa/wordpress/

Hello world!

Posted on March 28, 2013

Welcome to **MaheshaBSD Server Project**. Log in (username: **admin**, password: **manasa11**) to WordPress and start your **turnkey** Intranet/Internet blog right now – go to (*Posts*) *All Posts* and choose *Hello World* to edit this page. Add pictures and let all see them later in your posts.

This thing runs on OpenBSD 5.2. **The main goal of MaheshaBSD Server is to provide users with an easy to use turnkey Intranet and home server (and turnkey education)**. You can, too, open it to the world. Edit your *hosts* file in Unix (/etc/hosts), its content does not differ from Windows (you will find it in C:\winnt\system32\drivers\etc) – just add the following line in it and replace 192.168.1.200 with your actual local/public IP (to get your public URL, see whatismyip.com): **192.168.1.200 manasa** - FTP server will be then available at the following link: ftp://manasa, where you will find, too, the documentation for this thing. To check this web server's functionality, click on this link. Manage users from inside your Admin account (*Users > Add New*). When you allow registration of users in *Settings* via e-mail, password e-mailing will not work. It will be better to manage users under your strict administrator's eyes. **Visit my website and donate! Or click here.**

RECENT POSTS
- Hello world!

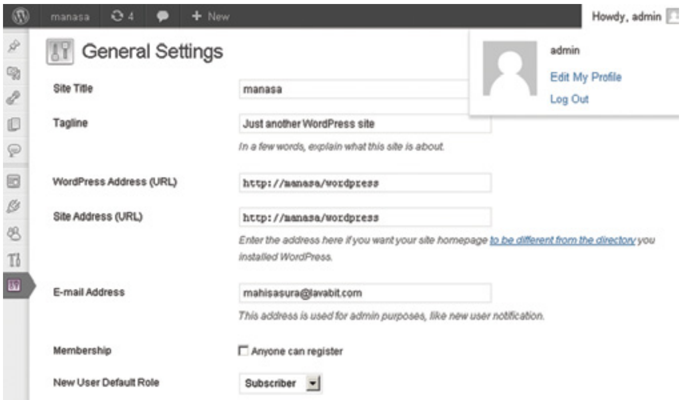
RECENT COMMENTS
- Anonymous on Hello world!
- Mr WordPress on Hello world!

ARCHIVES
- March 2013

CATEGORIES
- Uncategorized

META
- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Figure 5. *The WordPress introductory screen in MaheshaOpenBSD Server is self-explanatory*



General Settings

admin

Site Title: manasa

Tagline: Just another WordPress site
In a few words, explain what this site is about.

WordPress Address (URL): `http://manasa/wordpress`

Site Address (URL): `http://manasa/wordpress`
Enter the address here if you want your site homepage to be different from the directory you installed WordPress.

E-mail Address: mahisasura@lavabit.com
This address is used for admin purposes, like new user notification.

Membership: □ Anyone can register

New User Default Role: Subscriber

Figure 6. *It is advisable not to use IP's because they change*

MaheshaOpenBSD Server has a secure search engine (*https://startpage.com*). After you open Firefox or Seamonkey, you will see the picture of a white cobra. On other tabs, you will see some helpful sites. phpMyAdmin is available too (Figure 7).

When you make a DVD with the makeiso script, ignore the error message "UNEXPECTED INCONSISTENCY" after you boot MaheshaOpenBSD DVD. Just type exit and go. Everything will work just like with a USB flash drive.

After copying the files you want to share with your friends or installing anything necessary for you, run the script `/root/bin/sd0ro` to make MaheshaOpenBSD Server read-only (except for the directories that are in memory). This is a good security tip.

Change passwords!!! This is important as the passwords are publicly available. You may also change passwords when this is on a read-only USB flash drive (temporarily) as `/etc` resides in memory. To work with your new passwords on the next reboot, copy `/etc` to `/etc2` when this is writable (`/etc2` is not in memory).

Except for `/etc`, you can make all changes permanent when you next reboot (in `/var`, `/home`, `/root`) by running the script `/mfs/makeetcnomnt` from within MaheshaOpenBSD Server, or `/mfs/makeetc` if you mounted this USB im-

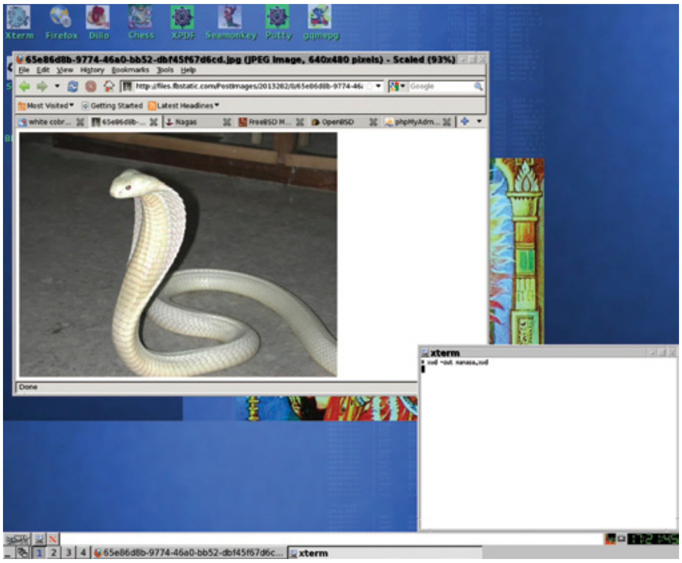


Figure 7. *The white cobra is a very rare cobra and it lives in the weirdest places of India*

age from OpenBSD running on your hard drive. The same applies to `/var` (makevar and makevarnomnt), `/root`, and the `/home` directories.

If you need a larger FTP storage, all you need to do is mount another disk to `/home/vsftpd/ftp`. MaheshaOpenBSD will work with any external or additional USB flash/hard drive.

Security

MaheshaOpenBSD Server is primarily an Intranet server, but if you plan to open this thing to the world, you must take a few security measures:

- change the root password
- change the MySQL password
- change the WordPress password
- kill VSFTPD (or MySQL) if you do not plan to use it
- upgrade WordPress

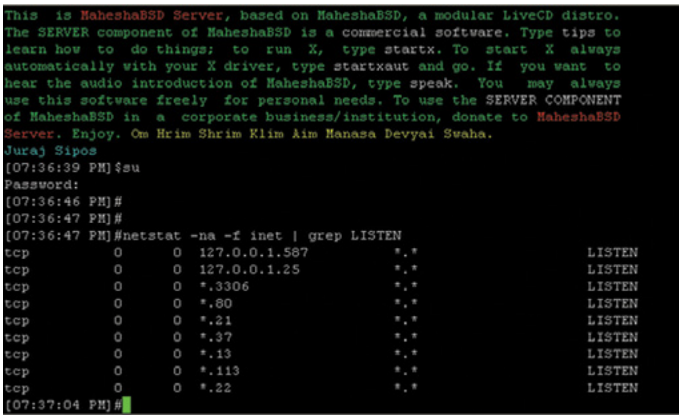
For WordPress, see the file `/var/www/htdocs/wordpress/wp.config.php`, where you will also find the MySQL password. There are many sites that will tell you how to change the MySQL password (Figure 8).

Logs are in `/var` – for example, `/var/logs/vsftpd.log` stores all the activity for the VSFTPD Server. Apache logs are in `/var/www/logs`; MySQL logs are in `/var/mysql`.

The best security tip is not to use SSH. To solve this, log in to your MaheshaOpenBSD box and look at the sample `/root/bin/atme` script. This script will turn the SSHD daemon off at a specific time and turn it on at the time you specify. This means that the best security measure is to log in to MaheshaOpenBSD over network, edit the `/root/bin/atme` script as you wish (the file contains the explanation in a commented text) and run it. The `/root/bin/atme` script will shut the SSHD daemon in two minutes and will start it at any time you specify. You do not need to edit anything else for this script to work. All you need to do is edit the `/root/bin/atme` script as follows:

```
at -f /root/bin/atme1 now + 2 minutes
- this will kill the SSHD daemon in 2 minutes
```

`sleep 3 ; at -f /root/bin/atme2 now + 5 minutes` – this will start the SSHD daemon in 5 minutes, but it is better to change the above value to a couple of hours or days (5 days). Only you will know when to log in. The above tip is my own invention and I use it with my FreeBSD box.



```
This  is MaheshaBSD Server, based on MaheshaBSD, a modular LiveCD distro.
The SERVER component of MaheshaBSD is a commercial software. Type tips to
learn how  to  do things;  to  run  X,  type startx. To  start  X  always
automatically with your X driver, type startxaut and go. If  you want  to
hear the audio introduction of MaheshaBSD, type speak.  You   may  always
use this software freely  for personal needs. To use the SERVER COMPONENT
of MaheshaBSD in  a  corporate business/institution, donate to MaheshaBSD
Server. Enjoy. Om Hrim Shrim Klim Aim Manasa Devyai Swaha.
Juraj Sipos
[07:36:39 PM]$su
Password:
[07:36:46 PM]#
[07:36:47 PM]#
[07:36:47 PM]#netstat -na -f inet | grep LISTEN
tcp          0       0  127.0.0.1.587          *.*                    LISTEN
tcp          0       0  127.0.0.1.25           *.*                    LISTEN
tcp          0       0  *.3306                 *.*                    LISTEN
tcp          0       0  *.80                   *.*                    LISTEN
tcp          0       0  *.21                   *.*                    LISTEN
tcp          0       0  *.37                   *.*                    LISTEN
tcp          0       0  *.13                   *.*                    LISTEN
tcp          0       0  *.113                  *.*                    LISTEN
tcp          0       0  *.22                   *.*                    LISTEN
[07:37:04 PM]#
```

Figure 8. *The following tcp/ip ports are open in the "default" installation*

Acknowledgments

I thank *www.rootbsd.net* for allowing me to distribute MaheshaBSD.

The fact that the user runs OpenBSD in MaheshaOpenBSD does not mean that the MaheshaOpenBSD Server is secure. We as humans must follow certain rules. Without them, we will become vulnerable. But MaheshaOpenBSD is on a writable USB flash drive and you are advised to secure it to your liking. Do keep in mind that users have different priorities. If security is your priority, change the secure level "-1" in `/etc/rc.securelevel` to higher numbers and take some other measures, too, like avoid running X. Use your own judgement!

Conclusion

Portability is a great feature of this thing! It is not easy to carry your computer with you, nor is it always effortless to edit configuration files on a computer that you do not own. MaheshaBSD Server or MaheshaOpenBSD Server is "a computer in your pocket", a difficult-to-find thing for thieves who will always find it easier to steal notebooks from cars or cafeteria tables. OpenBSD is one of a few projects made with love and great enthusiasm. I hope that this great work, MaheshaOpenBSD, will help advocate for OpenBSD and attract people outside of the BSD world.

JURAJ SIPOS

Juraj lives in Slovakia and he works in a library within an educational institute. Some time in the past, he was fortunate to travel around the world and he spent a bit of time in India and Australia. Juraj's hobbies are computers, mostly Unix, but spirituality too. His first published computer article was Xmodmap Howto (http://tldp.org/HOWTO/Intkeyb/). In addition to computers, he is very interested in Hinduism but not really the guru side of things, but more-so freedom and Tantric self-actualization. More at his website: http://www.freebsd.nfo.sk/ (FreeBSD) http://www.freebsd.nfo.sk/maheshaeng.htm (MaheshaBSD)

Security Policy Development in

TrustedBSD MAC Framework

Trusted Operating Systems are the next level of system security. They offer both new security features and a high assurance of successful implementation. Trusted systems differ from secure systems in many principles. Trusted Systems established the concept of "ranking" systems with different degrees of trustworthiness.

What you will learn…

- How to write OS Security policy
- MAC Security Framework structure
- Installation and configuration of Nagios with MAC Security Framework

What you should know…

- Basic understanding of FreeBSD
- How to install FreeBSD
- How to install port (application) on FreeBSD

In such systems, users decide on trustworthiness and make a judgment based on the security of systems. Operating systems have to implement security policies, and different mechanisms are used to enforce such policies. There are various operating system security policies such as the Massachusetts Library System (MLS) and Biba, which was developed by Kenneth J. Biba. This article describes the overall process of developing and applying different security policies within the FreeBSD kernel under the TrustedBSD MAC Security Framework.

Introduction

A kernel is a central component of an operating system. It acts as an interface between user applications and hardware. The main purpose of the kernel is to manage the communication between the software (user level applications) and the hardware (CPU, disk memory, etc). The main kernel tasks are: Process Management, Device Management, Memory Management, Interrupt Handling, I/O Communication and File System. New kernel structures arose consisting of several modules classified into static (base) kernel modules and dynamic "pluggable" kernel modules. The main advantage of dynamic kernel modules is the ability to be attached at run-time in a flexible way. In the static version of the kernel, all modules must be present at kernel compilation time. On the other hand, in the dynamic scenario, modules have to be available only during run-time. Furthermore, the security and performance tradeoffs are formulating the best approach to use in kernel compilation. On the other hand, different security models and policies were implemented in OS kernels to add many security features to the "plain" kernels. From an operating system point of view, security policies are the restrictions that administrators would like to apply while mechanisms are the procedures used to enforce such policies. There are various operating system security policies such as MLS and Biba policies.

Security Policies Problems

The variety and non-standardization of the design of security policies introduced conflicts between implementations of these policies. Different vendor implementations along with security policies' intentions caused big headaches for security developers. In addition, frequent changes in user requirements and the need for policy customizations pushed security developers to seek a new security methodology for covering these issues. In the past, kernels were adapted to one security model at most. Furthermore, we had to include the security policy in the kernel configuration file before compiling the kernel. However, this inflexibility in embedding security policies in the kernel was a huge problem. Continuous work by researchers led to the introduction of new concepts for OS security. This technique focuses on implementing an intermediate layer between security models and kernel services. This intermediate layer offers policy composition manipulation and customization in addition to the capability to attach different security policies at run-time. One of the most common security frameworks implementing the mentioned points is the MAC framework. The MAC framework provides a set of wrappers for use by different policies' vendors. Figure 1 describes the overall architecture of the MAC framework.

MAC Security Framework in detail

Two of the most significant security mechanisms are file system *Access Control Lists* (ACLs) and *Mandatory Access Control* (MAC) facilities. Mandatory Access Control allows new access control modules to be loaded to support new security policies. Some modules provide protections of a narrow subset of the system, hardening a particular service. Others provide comprehensive labeled security across all subjects and objects. The mandatory part of the definition comes from the fact that the enforcement of the controls is done by administrators and the system and is not left up to the discretion of users as is done with discretionary access control (DAC, the standard file and System V IPC permissions on FreeBSD). One of the most common implementations of the MAC Security is the TrustedBSD MAC Framework. The TrustedBSD MAC framework provides a mechanism to allow the compile-time or run-time extension of the kernel access control model. New system policies may be implemented as kernel modules and linked to the kernel; if multiple policy modules are present, their results will be composed. The MAC Framework provides a variety of access control infrastructure services to assist policy writers, including support for transient and persistent policy-agnostic object security labels.

Detailed Architecture

MAC Framework Interfaces for Kernel Services

The MAC Framework presents a set of entry points to selected kernel services, permitting the services to provide event notification to the MAC framework and providing the ability for the MAC Framework to maintain a security label within kernel objects maintained by the kernel services. In FreeBSD, the interface used by its kernel services to communicate with the MAC Framework is defined in sys/mac.h. This includes the APIs for all entry points from the kernel services. In addition, `sys/ label.h` defines struct label, a data structure used to store policy-agnostic label data in kernel objects. This structure is embedded into many kernel service structures.

Framework Kernel Service Entry Points

Modifications have been made to kernel services to invoke MAC Framework entry points. These modifications affect object initialization, association/creation, and destruction, as well as in common paths requiring access control at high levels in the kernel. With layered services, it is often necessary to defer access control decisions until enough information is available.

Framework Implementation

Entry point implementations, label primitives, policy registration, and user/kernel APIs are centralized in `kern_mac.c`.

Framework Interface for Policies

The MAC Framework provides several interfaces to security policy implementations, including interfaces for policy management, label storage, process label management, object life cycle, access control, and system life cycle. Extensions implement arbitrary subsets of the available interfaces, allowing implementers to select the events and services that are relevant to a particular policy. Interfaces common to the framework and policies and defined in `sys/mac policy.h`. Definitions include entry point and registration interfaces, as well as common access methods for MAC Framework services.

Policy Implementations

Each policy is represented by one kernel module, discouraging inter-dependency. Typical policies are implemented in a single C file, but complex policies are implemented over many files.

Interfaces to User Processes

Interfaces for user processes are defined in `sys/mac.h`, implemented in libc, and may be dynamically linked into any applications.

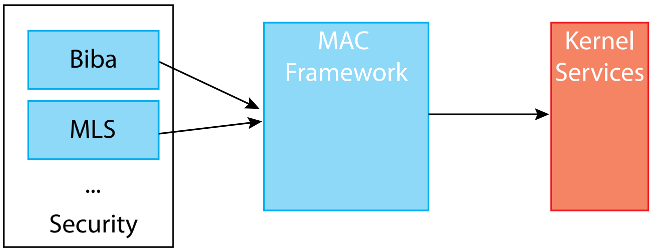


Figure 1. *MAC Framework Overall Architecture*

Main Idea

The main idea behind TrustedBSD MAC framework is labeling different kernel objects to provide the ability to track them. A label is a security attribute which can be applied to files, directories, or other items in the system. It could be considered a confidentiality stamp; when a label is placed on a file, it describes the security properties for that specific file and will only permit access by files, users, resources, etc. with a similar security setting. The meaning and interpretation of label values depends on the policy configuration. While some policies might treat a label as representing the integrity or secrecy of an object, other policies might use labels to hold rules for access. Table 1 shows the basic kernel elements that are labeled for kernel elements.

Table 1. *Labeled Objects*

Structure	Description
struct ucred	Process credential
struct vnode	VFS node
struct socket	BSD IPC socket
struct pipe	IPC pipe
struct mbuf	In-flight datagram
struct mount	File system mount
struct ifnet	Network interface
struct devfs_dirent	Devfs entry
struct ipq	IP fragment queue
struct bpf_desc	BPF packet sniff device

Now, let's move to MAC implementation in FreeBSD kernels. The following points clarify the required steps to allow run-time MAC security on your FreeBSD machine:

Adding MAC support to the kernel

Kernels should have MAC support to give the flexibility of implementing and composing security policies. To achieve this, add mac option to your kernel configuration file which is `/usr/src/sys/conf/GENERIC` by default.

```
options MAC
```

Next, recompile the kernel to rephrase the kernel for MAC support. In order to compile a GENERIC kernel, do the following steps:

- Change to the `/usr/src` directory:

```
# cd /usr/src
```

- Compile the kernel:

```
# make buildkernel KERNCONF=GENERIC
```

Note: You can omit `KERNCONF=GENERIC` if you are referring to the default configuration file.

- Install the new kernel:

```
# make installkernel KERNCONF=GENERIC
```

Note: You can omit `KERNCONF=GENERIC` if you are referring to the default configuration file.

Embedding MAC Policy

Security policies are either linked directly into the kernel, or compiled into loadable kernel modules that may be loaded at boot, or dynamically using the module loading system calls at runtime.

Policy Declaration

Modules may be declared using the `MAC_POLICY_SET()` macro which names the policy, provides a reference to the MAC entry point vector, provides load-time flags determining how the policy framework should handle the policy, and optionally requests the allocation of label state by the framework (Listing 1).

The MAC policy entry point vector, `mac_policy_ops` in this example, associates functions defined in the module with specific entry points. Of specific interest during module registration are the `.mpo_destroy` and `.mpo_init` entry points. `.mpo_init` will be invoked once a policy is successfully registered with the module framework but prior to any

Listing 1. *mac_policy_ops function*

```
static struct mac_policy_ops mac_policy_ops =
{
        .mpo_destroy = mac_policy_destroy,
        .mpo_init = mac_policy_init,
        .mpo_init_bpfdesc_label = mac_policy_init_
                   bpfdesc_label,
        .mpo_init_cred_label = mac_policy_init_label,
/* ... */
        .mpo_check_vnode_setutimes = mac_policy_check_
                   vnode_setutimes,
        .mpo_check_vnode_stat = mac_policy_check_
                   vnode_stat,
        .mpo_check_vnode_write = mac_policy_check_
                   vnode_write,
};
```

other entry points becoming active. This permits the policy to perform any policy-specific allocation and initialization, such as initialization of any data or locks. `.mpo_destroy` will be invoked when a policy module is unloaded to permit releasing of any allocated memory and destruction of locks. Currently, these two entry points are invoked with the MAC policy list mutex held to prevent any other entry points from being invoked. This will be changed, but in the meantime, policies should be careful about what kernel primitives they invoke to avoid lock ordering or sleeping problems.

The policy declaration's module name field exists so that the module may be uniquely identified for the purposes of module dependencies. An appropriate string should be selected. The full string name of the policy is displayed to the user via the kernel log during loading and unloading events, and also exported when providing status information to user processes.

Label Configuration

Virtually all aspects of label policy module configuration will be performed using the base system utilities. These commands provide a simple interface for object or subject configuration or the manipulation and verification of the configuration.

All configuration may be done by use of the `setfmac(8)` and `setpmac(8)` utilities. The `setfmac` command is used to set `MAC` labels on system objects while the `setpmac` command is used to set the labels on system subjects. Observe:

```
# setfmac biba/high test
```

If no errors occurred with the command above, a prompt will be returned. The only time these commands are not quiescent is when an error occurred, similar to the `chmod(1)` and `chown(8)` commands. In some cases, this error may be a `Permission denied` and is usually obtained when the label is being set or modified on an object which is restricted. The system administrator may use the following commands to overcome this:

```
# setfmac biba/high test
```

This is usually what you get when you try to set MAC label to "test" directly:

```
"Permission denied" !!!
```

To avoid this problem, wrap label change into new mac process. The next command forks new mac process and assigns mac label with highest possible privileges

```
# setpmac biba/low setfmac biba/high test
```

Finally, verify what you just did!

```
# getfmac test
test: biba/high
```

Two types of labels are available: singlelabel and multilabel. By default, all the labels are singlelabel items. The `multilabel` option permits each subject or object to have its own independent MAC label in place of standard `singlelabel` option which enforces single label throughout the partition. The `multilabel` and `singlelabel` options are only required for the policies implementing label feature, including the Biba, Low Water-Mark mandatory access control (Lomac), MLS and Security Enhanced BSD (SEBSD) policies. The multilabel is required when we have different policies implemented to set a policy for each labeled object. To set multilabel on the filesystem:

```
# tunefs -l enable /
```

Policy Configuration

The Table 2 shows the set of predefined policies for TrustedBSD MAC for use instead of creating new ones.

Table 2. *MAC Policies*

Policy	Description
mac_biba	Hierarchal fixed-label integrity
mac_bsdextended	"File system firewall" using existing credentials/permissions
mac_ifoff	Interface silencing
mac_lomac	Hierarchal floating-label integrity
mac_mls	Multi-Level Security with compartments
mac_none	Prototype stub policy
mac_partition	Inter-process visibility policy based on process partition labels
mac_seeotheruids	Inter-process visibility policy based on existing credentials.
mac_test	MAC Framework invariant tests
sebsd	Port of the SELinux/FLASK/TE

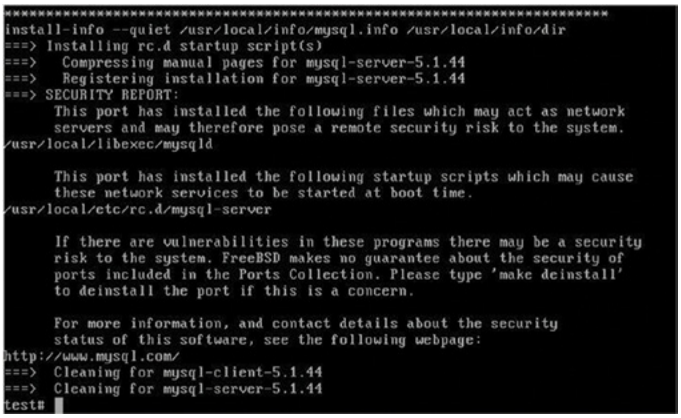
Case Study (Apply MAC to files and Test it via Nagios)

In this example, we will simulate a typical MAC Security scenario for Sendmail service with the aid of Nagios, one of the most common applications in the field of IT infrastructure monitoring. I highly recommend trying this example on a virtual machine because failing in administering your system well might cause user account inaccessibility. Before starting this example, the `multilabel` option must be set on each file system. Missing this label will result in errors.

- Setting multilabel flag over the filesystem through single user mode:

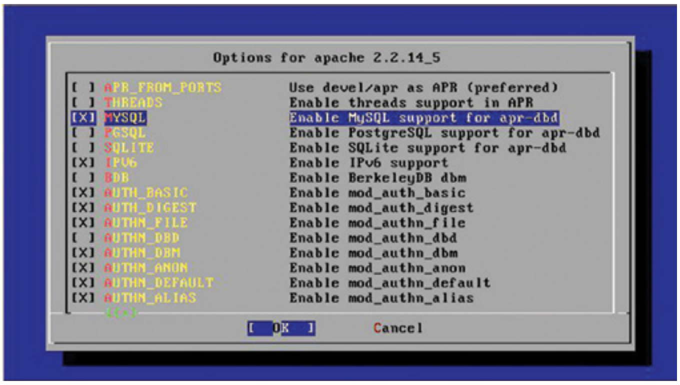
```
# tunefs -l enable /
```

- Install MySQL Server



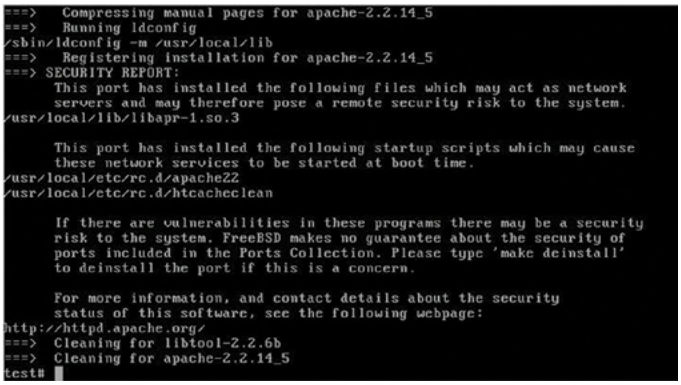
```
****************************************************************************
install-info --quiet /usr/local/info/mysql.info /usr/local/info/dir
===> Installing rc.d startup script(s)
===>   Compressing manual pages for mysql-server-5.1.44
===>   Registering installation for mysql-server-5.1.44
===> SECURITY REPORT:
      This port has installed the following files which may act as network
      servers and may therefore pose a remote security risk to the system.
/usr/local/libexec/mysqld

      This port has installed the following startup scripts which may cause
      these network services to be started at boot time.
/usr/local/etc/rc.d/mysql-server

      If there are vulnerabilities in these programs there may be a security
      risk to the system. FreeBSD makes no guarantee about the security of
      ports included in the Ports Collection. Please type 'make deinstall'
      to deinstall the port if this is a concern.

      For more information, and contact details about the security
      status of this software, see the following webpage:
http://www.mysql.com/
===>  Cleaning for mysql-client-5.1.44
===>  Cleaning for mysql-server-5.1.44
test#
```

Figure 2. *MySQL Server after installation*



```
                      Options for apache 2.2.14_5

[ ] APR_FROM_PORTS          Use devel/apr as APR (preferred)
[ ] THREADS                 Enable threads support in APR
[X] MYSQL                   Enable MySQL support for apr-dbd
[ ] PGSQL                   Enable PostgreSQL support for apr-dbd
[ ] SQLITE                  Enable SQLite support for apr-dbd
[X] IPV6                    Enable IPv6 support
[ ] BDB                     Enable BerkeleyDB dbm
[X] AUTH_BASIC              Enable mod_auth_basic
[X] AUTH_DIGEST             Enable mod_auth_digest
[X] AUTHN_FILE              Enable mod_authn_file
[ ] AUTHN_DBD               Enable mod_authn_dbd
[X] AUTHN_DBM               Enable mod_authn_dbm
[X] AUTHN_ANON              Enable mod_authn_anon
[X] AUTHN_DEFAULT           Enable mod_authn_default
[X] AUTHN_ALIAS             Enable mod_authn_alias
    v(+)

                  [  OK  ]        Cancel
```

Figure 3. *Apache Configuration Parameters*



```
===>   Compressing manual pages for apache-2.2.14_5
===>   Running ldconfig
/sbin/ldconfig -m /usr/local/lib
===>   Registering installation for apache-2.2.14_5
===> SECURITY REPORT:
      This port has installed the following files which may act as network
      servers and may therefore pose a remote security risk to the system.
/usr/local/lib/libapr-1.so.3

      This port has installed the following startup scripts which may cause
      these network services to be started at boot time.
/usr/local/etc/rc.d/apache22
/usr/local/etc/rc.d/htcacheclean

      If there are vulnerabilities in these programs there may be a security
      risk to the system. FreeBSD makes no guarantee about the security of
      ports included in the Ports Collection. Please type 'make deinstall'
      to deinstall the port if this is a concern.

      For more information, and contact details about the security
      status of this software, see the following webpage:
http://httpd.apache.org/
===>  Cleaning for libtool-2.2.6b
===>  Cleaning for apache-2.2.14_5
test#
```

Figure 4. *Apache after installation*

```
# cd /usr/ports/databases/mysql51-server/
# make install clean
```

Note: this port, i.e. application, will install the MySQL server and the MySQL client as well (Figure 2).

- Enable MySQL service at system startup

```
#echo 'mysql_enable="YES"' >> /etc/rc.conf
```

- Start MySQL service immediately to avoid rebooting your machine

```
# /usr/local/etc/rc.d/mysql-server start
Starting mysql.
#
```

- Change Administrator Password for MySQL

```
# mysqladmin -u root password adminpassword
```

- Install Apache Application Server

```
# cd /usr/ports/www/apache22/
# make install clean
```

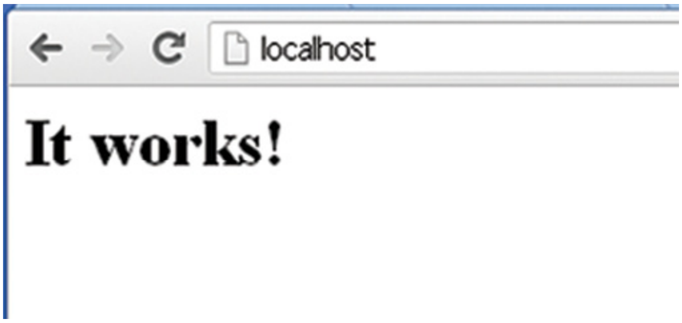
Note: Make sure that you enabled mysql-support (Figure 3 and Figure 4)

- Enable Apache service at System Startup

```
# echo 'apache22_enable="YES"' >> /etc/rc.conf
```

Listing 2. *Start Apache Service*

```
# echo 'apache22_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/apache22 start
Performing sanity check on apache22 configuration:
Syntax OK
Starting apache22.
#
```



It works!

Figure 5. *Testing Apache*

- Start Apache Service Immediately to avoid rebooting your system (Listing 2)
- Check if your Apache server is up through http://yourIP/ (Figure 5)
- Install PHP5

```
# cd /usr/ports/lang/php5/
# make install clean
```

Note: Make sure that you enabled apache support (Figure 6)

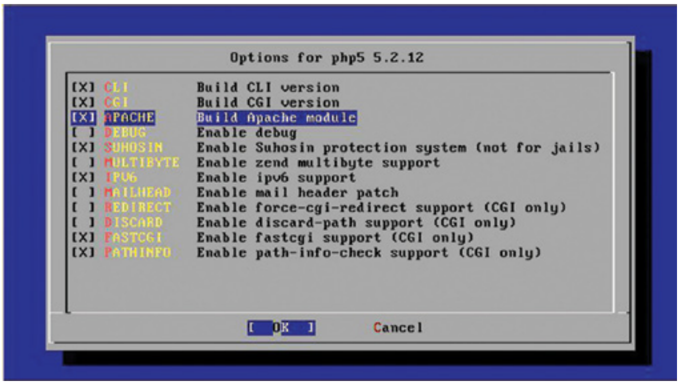
- Install PHP5 Extensions

```
# cd /usr/ports/lang/php5-extensions/
# make install clean
```

Note: Make sure that you enabled mysql-support and session support (Figure 7)

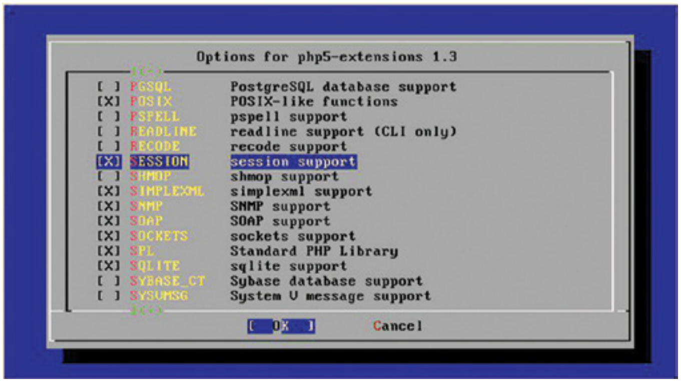
Listing 3. *Enable PHP in Apache*

```
DirectoryIndex index.php index.html index.htm
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```



```
                    Options for php5 5.2.12

 [X] CLI          Build CLI version
 [X] CGI          Build CGI version
 [X] APACHE       Build Apache module
 [ ] DEBUG        Enable debug
 [X] SUHOSIN      Enable Suhosin protection system (not for jails)
 [ ] MULTIBYTE    Enable zend multibyte support
 [X] IPV6         Enable ipv6 support
 [ ] MAILHEAD     Enable mail header patch
 [ ] REDIRECT     Enable force-cgi-redirect support (CGI only)
 [ ] DISCARD      Enable discard-path support (CGI only)
 [X] FASTCGI      Enable fastcgi support (CGI only)
 [X] PATHINFO     Enable path-info-check support (CGI only)
```

Figure 6. *PHP5 Configuration Parameters*



```
                 Options for php5-extensions 1.3
        ^(-)
    [ ] PGSQL        PostgreSQL database support
    [X] POSIX        POSIX-like functions
    [ ] PSPELL       pspell support
    [ ] READLINE     readline support (CLI only)
    [ ] RECODE       recode support
    [X] SESSION      session support
    [ ] SHMOP        shmop support
    [X] SIMPLEXML    simplexml support
    [X] SNMP         SNMP support
    [X] SOAP         SOAP support
    [X] SOCKETS      sockets support
    [X] SPL          Standard PHP Library
    [X] SQLITE       sqlite support
    [ ] SYBASE_CT    Sybase database support
    [ ] SYSVMSG      System V message support
        v(+)
```

Figure 7. *PHP5 Extensions Configuration Parameters*

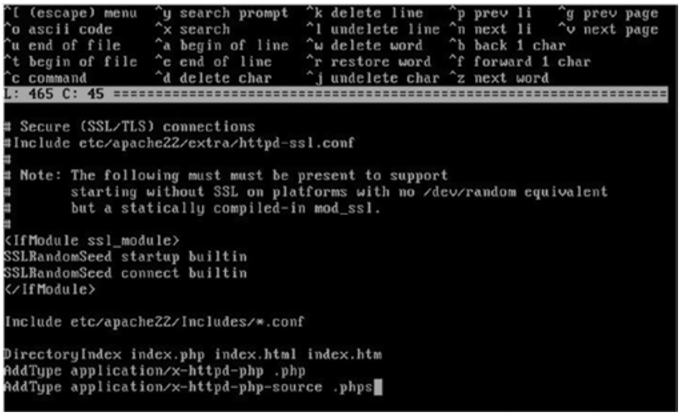
- Rename php.ini-dist to php.ini

```
# cp usr/local/etc/php.ini-dist /usr/local/etc/php.ini
```

- Configure Apache to enable PHP support and append the lines from Listing 3 to apache configuration file (`httpd.conf`) (Figure 8)

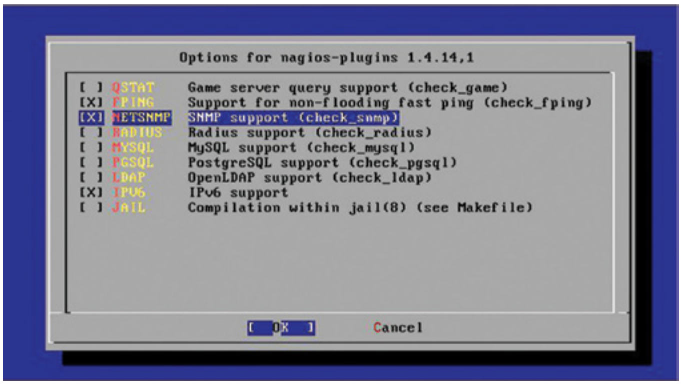
```
# ee /usr/local/etc/apache22/httpd.conf
```

- Install Nagios



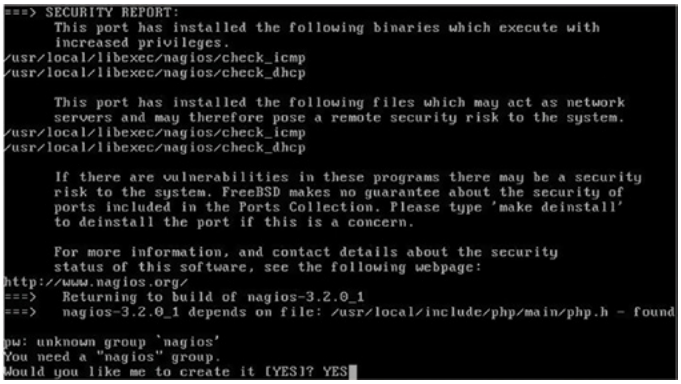
```
^[ (escape) menu  ^y search prompt  ^k delete line   ^p prev li   ^g prev page
^o ascii code     ^x search         ^l undelete line ^n next li   ^v next page
^u end of file    ^a begin of line  ^w delete word   ^b back 1 char
^t begin of file  ^e end of line    ^r restore word  ^f forward 1 char
^c command        ^d delete char    ^j undelete char ^z next word
L: 465 C: 45 ==================================================================

# Secure (SSL/TLS) connections
#Include etc/apache22/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

Include etc/apache22/Includes/*.conf

DirectoryIndex index.php index.html index.htm
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Figure 8. *Append Listing 3 to the end of the file*



```
              Options for nagios-plugins 1.4.14,1

 [ ] QSTAT      Game server query support (check_game)
 [X] FPING      Support for non-flooding fast ping (check_fping)
 [X] NETSNMP    SNMP support (check_snmp)
 [ ] RADIUS     Radius support (check_radius)
 [ ] MYSQL      MySQL support (check_mysql)
 [ ] PGSQL      PostgreSQL support (check_pgsql)
 [ ] LDAP       OpenLDAP support (check_ldap)
 [X] IPV6       IPv6 support
 [ ] JAIL       Compilation within jail(8) (see Makefile)
```

Figure 9. *Enable SNMP in Nagios Configuration*



```
===> SECURITY REPORT:
      This port has installed the following binaries which execute with
      increased privileges.
/usr/local/libexec/nagios/check_icmp
/usr/local/libexec/nagios/check_dhcp

      This port has installed the following files which may act as network
      servers and may therefore pose a remote security risk to the system.
/usr/local/libexec/nagios/check_icmp
/usr/local/libexec/nagios/check_dhcp

      If there are vulnerabilities in these programs there may be a security
      risk to the system. FreeBSD makes no guarantee about the security of
      ports included in the Ports Collection. Please type 'make deinstall'
      to deinstall the port if this is a concern.

      For more information, and contact details about the security
      status of this software, see the following webpage:
http://www.nagios.org/
===>  Returning to build of nagios-3.2.0_1
===>  nagios-3.2.0_1 depends on file: /usr/local/include/php/main/php.h - found

pw: unknown group `nagios'
You need a "nagios" group.
Would you like me to create it [YES]? YES
```

Figure 10. *Nagios User and Group creation*

XRY
COMPLETE

IT'S NOT ABOUT DATA. IT'S ABOUT MEANING.

If you think mobile forensics is just about extracting data – think again. Its not only what you get, but what you do with it that really makes the difference.

XRY has an intuitive GUI that's easier to use, has better display capabilities and superior analysis functionality.

MICRO SYSTEMATION
msab.com

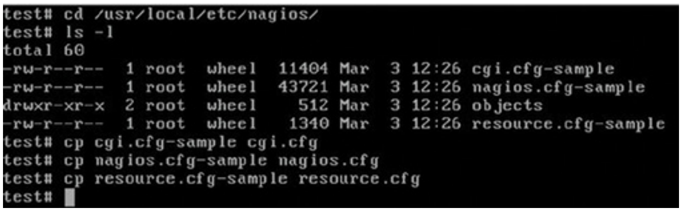
```
# cd /usr/ports/net-mgmt/nagios
# make install clean
```

Note: Make sure that you enabled SNMP support in Nagios-plugins installation which follows Nagios installation (Figure 9)

- Create Nagios User and Group (Figure 10)
- Enable Nagios service at system startup

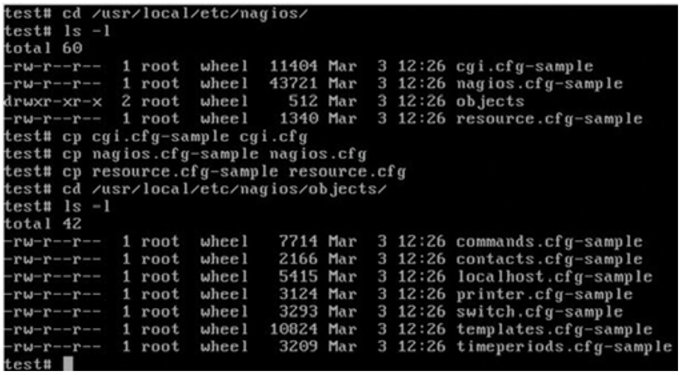
```
# echo 'nagios_enable="YES" >> /etc/rc.conf
```

- Configure Nagios (Figure 11-13)
- Start Nagios service immediately to avoid system rebooting (Figure 14)
- Configure Apache to support Nagios and append the lines from Listing 4 to `/usr/local/etc/apache22/httpd.conf`



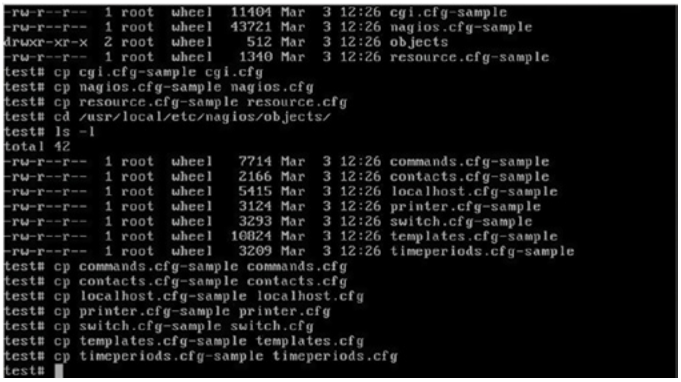
```
test# cd /usr/local/etc/nagios/
test# ls -l
total 60
-rw-r--r--  1 root  wheel  11404 Mar  3 12:26 cgi.cfg-sample
-rw-r--r--  1 root  wheel  43721 Mar  3 12:26 nagios.cfg-sample
drwxr-xr-x  2 root  wheel    512 Mar  3 12:26 objects
-rw-r--r--  1 root  wheel   1340 Mar  3 12:26 resource.cfg-sample
test# cp cgi.cfg-sample cgi.cfg
test# cp nagios.cfg-sample nagios.cfg
test# cp resource.cfg-sample resource.cfg
test# 
```

Figure 11. *Nagios Post Installation Initial Configuration*



```
test# cd /usr/local/etc/nagios/
test# ls -l
total 60
-rw-r--r--  1 root  wheel  11404 Mar  3 12:26 cgi.cfg-sample
-rw-r--r--  1 root  wheel  43721 Mar  3 12:26 nagios.cfg-sample
drwxr-xr-x  2 root  wheel    512 Mar  3 12:26 objects
-rw-r--r--  1 root  wheel   1340 Mar  3 12:26 resource.cfg-sample
test# cp cgi.cfg-sample cgi.cfg
test# cp nagios.cfg-sample nagios.cfg
test# cp resource.cfg-sample resource.cfg
test# cd /usr/local/etc/nagios/objects/
test# ls -l
total 42
-rw-r--r--  1 root  wheel   7714 Mar  3 12:26 commands.cfg-sample
-rw-r--r--  1 root  wheel   2166 Mar  3 12:26 contacts.cfg-sample
-rw-r--r--  1 root  wheel   5415 Mar  3 12:26 localhost.cfg-sample
-rw-r--r--  1 root  wheel   3124 Mar  3 12:26 printer.cfg-sample
-rw-r--r--  1 root  wheel   3293 Mar  3 12:26 switch.cfg-sample
-rw-r--r--  1 root  wheel  10824 Mar  3 12:26 templates.cfg-sample
-rw-r--r--  1 root  wheel   3209 Mar  3 12:26 timeperiods.cfg-sample
test# 
```

Figure 12. *Nagios Post Initial Configuration Verification*

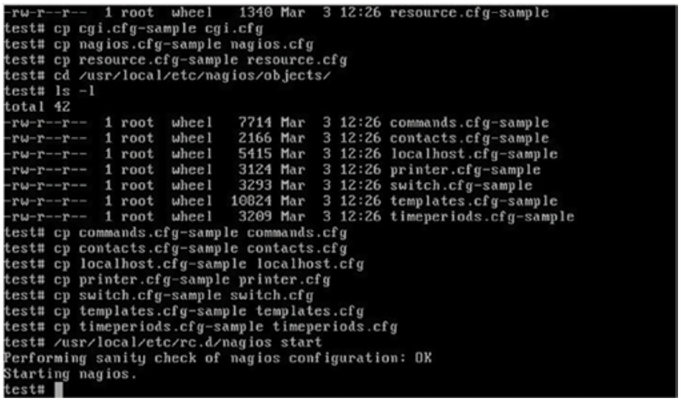


```
-rw-r--r--  1 root  wheel  11404 Mar  3 12:26 cgi.cfg-sample
-rw-r--r--  1 root  wheel  43721 Mar  3 12:26 nagios.cfg-sample
drwxr-xr-x  2 root  wheel    512 Mar  3 12:26 objects
-rw-r--r--  1 root  wheel   1340 Mar  3 12:26 resource.cfg-sample
test# cp cgi.cfg-sample cgi.cfg
test# cp nagios.cfg-sample nagios.cfg
test# cp resource.cfg-sample resource.cfg
test# cd /usr/local/etc/nagios/objects/
test# ls -l
total 42
-rw-r--r--  1 root  wheel   7714 Mar  3 12:26 commands.cfg-sample
-rw-r--r--  1 root  wheel   2166 Mar  3 12:26 contacts.cfg-sample
-rw-r--r--  1 root  wheel   5415 Mar  3 12:26 localhost.cfg-sample
-rw-r--r--  1 root  wheel   3124 Mar  3 12:26 printer.cfg-sample
-rw-r--r--  1 root  wheel   3293 Mar  3 12:26 switch.cfg-sample
-rw-r--r--  1 root  wheel  10824 Mar  3 12:26 templates.cfg-sample
-rw-r--r--  1 root  wheel   3209 Mar  3 12:26 timeperiods.cfg-sample
test# cp commands.cfg-sample commands.cfg
test# cp contacts.cfg-sample contacts.cfg
test# cp localhost.cfg-sample localhost.cfg
test# cp printer.cfg-sample printer.cfg
test# cp switch.cfg-sample switch.cfg
test# cp templates.cfg-sample templates.cfg
test# cp timeperiods.cfg-sample timeperiods.cfg
test# 
```

Figure 13. *More Configuration on Nagios*

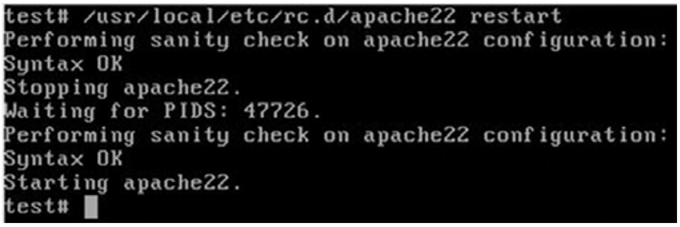
Listing 4. *Apache Configurations for Nagios*

```
ee /usr/local/etc/apache22/httpd.conf
ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/
Alias /nagios /usr/local/www/nagios/

<Directory /usr/local/www/nagios>
        Options None
        AllowOverride None
        Order allow,deny
        Allow from all
        AuthName "Nagios Access"
        AuthType Basic
       AuthUSerFile /usr/local/etc/nagios/htpasswd.users
        Require valid-user
</Directory>

<Directory /usr/local/www/nagios/cgi-bin>
        Options ExecCGI
        AllowOverride None
        Order allow,deny
        Allow from all
        AuthName "Nagios Access"
        AuthType Basic
       AuthUSerFile /usr/local/etc/nagios/htpasswd.users
        Require valid-user
</Directory>
```



```
-rw-r--r--  1 root  wheel   1340 Mar  3 12:26 resource.cfg-sample
test# cp cgi.cfg-sample cgi.cfg
test# cp nagios.cfg-sample nagios.cfg
test# cp resource.cfg-sample resource.cfg
test# cd /usr/local/etc/nagios/objects/
test# ls -l
total 42
-rw-r--r--  1 root  wheel   7714 Mar  3 12:26 commands.cfg-sample
-rw-r--r--  1 root  wheel   2166 Mar  3 12:26 contacts.cfg-sample
-rw-r--r--  1 root  wheel   5415 Mar  3 12:26 localhost.cfg-sample
-rw-r--r--  1 root  wheel   3124 Mar  3 12:26 printer.cfg-sample
-rw-r--r--  1 root  wheel   3293 Mar  3 12:26 switch.cfg-sample
-rw-r--r--  1 root  wheel  10824 Mar  3 12:26 templates.cfg-sample
-rw-r--r--  1 root  wheel   3209 Mar  3 12:26 timeperiods.cfg-sample
test# cp commands.cfg-sample commands.cfg
test# cp contacts.cfg-sample contacts.cfg
test# cp localhost.cfg-sample localhost.cfg
test# cp printer.cfg-sample printer.cfg
test# cp switch.cfg-sample switch.cfg
test# cp templates.cfg-sample templates.cfg
test# cp timeperiods.cfg-sample timeperiods.cfg
test# /usr/local/etc/rc.d/nagios start
Performing sanity check of nagios configuration: OK
Starting nagios.
test# 
```

Figure 14. *Starting Nagios*



```
test# /usr/local/etc/rc.d/apache22 restart
Performing sanity check on apache22 configuration:
Syntax OK
Stopping apache22.
Waiting for PIDS: 47726.
Performing sanity check on apache22 configuration:
Syntax OK
Starting apache22.
test# 
```

Figure 15. *Restarting Nagios*

- Restart Apache to apply changes (Figure 15)
- Login to Nagios (Figure 16 and Figure 17)
- Create an insecure User Class

Begin the procedure by adding the user class from Listing 5 to the `/etc/login.conf` file.

Finally, add the following line to the default user class:

```
:label=biba/high:      #Set privileges higher than other
                        labels
```

Once this is completed, the following command must be issued to rebuild the database:

```
# cap_mkdb /etc/login.conf
```

Boot Configuration

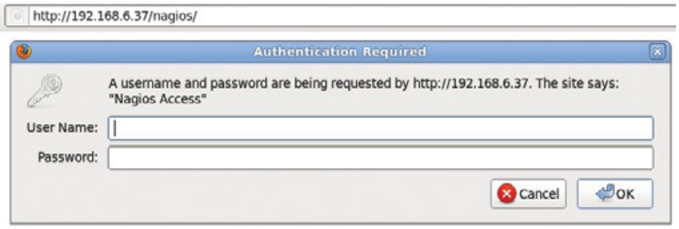
Next, we have to adapt boot configurations to load the MAC policies at boot time. Add the following lines to `/boot/loader.conf` so the required modules will load during system initialization:

```
mac_biba_load=”YES”
mac_seeotheruids_load=”YES”
```

Configure Users

Set the `root` user to the default class using:

```
# pw usermod root -L default
```



http://192.168.6.37/nagios/

Authentication Required

A username and password are being requested by http://192.168.6.37. The site says: "Nagios Access"

User Name:

Password:

Figure 16. *Logging into Nagios*

http://192.168.6.37/nagios/

Nagios®

Nagios Core Version 3.x Documentation

http://www.nagios.org

Copyright © 2009 Nagios Core Development Team and Community Contributors.
Copyright © 1999-2009 Ethan Galstad

Portions copyright by Nagios Community members. See the THANKS file for more information.

Last Updated: 06-16-2009

[Table of Contents]

Figure 17. *Nagios Homepage*

Now, all user accounts that are not `root` or system users will require a login class. Otherwise, users will be unable to execute common commands such as vi. The following `sh` script should do the trick:

```
# for x in `awk -F: ‘($3 >= 1001) && ($3 != 65534) { print $1 }’ \
/etc/passwd`; do pw usermod $x -L default; done;
```

Drop `nagios` and `www` users into the insecure class to avoid being caught by MAC:

```
# pw usermod nagios -L insecure
# pw usermod www -L insecure
```

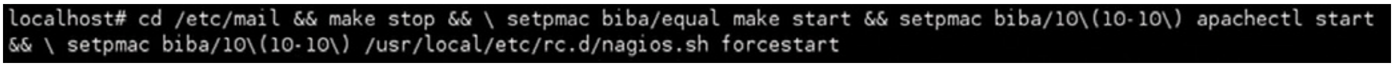
Create the Contexts File

A contexts file should now be created; the example file shown on Listing 6 should be placed in `/etc/policy.contexts`.

This policy enforces security by setting restrictions on the flow of information. In this specific configuration, users `root` and others should never be allowed to access Nagios. Configuration files and processes that are a part

Listing 5. *Nagios Configuration for MAC Security*

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/
                    sbin:/usr/local/bin
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=biba/10(10-10):     #Set privileges to grade of
                    10 with compartments 10 and 10
```



```
localhost# cd /etc/mail && make stop && \ setpmac biba/equal make start && setpmac biba/10\(10-10\) apachectl start
&& \ setpmac biba/10\(10-10\) /usr/local/etc/rc.d/nagios.sh forcestart
```

Figure 18. *Testing MAC on Nagios, Apache and Sendmail*

of Nagios will be completely self contained or jailed. This file may be read into our system by issuing the following command:

```
# setfsmac -ef /etc/policy.contexts /
```

The previous command sets Biba policies to all the proposed objects.

Listing 6. *Policy Configuration for MAC, Nagios & Apache*

```
# This is the default BIBA policy for this system.
# System:
/var/run                        biba/equal       #Set
                   privileges equal to other labels
/var/run/*                      biba/equal

/dev                            biba/equal
/dev/*                          biba/equal

/var             biba/equal
/var/spool                      biba/equal
/var/spool/*                    biba/equal

/var/log                        biba/equal
/var/log/*                      biba/equal

/tmp             biba/equal
/tmp/*           biba/equal
/var/tmp         biba/equal
/var/tmp/*          biba/equal

/var/spool/mqueue      biba/equal
/var/spool/clientmqueue      biba/equal

# For Nagios:
/usr/local/etc/nagios
/usr/local/etc/nagios/*         biba/10

/var/spool/nagios               biba/10
/var/spool/nagios/*             biba/10

# For apache
/usr/local/etc/apache           biba/10
/usr/local/etc/apache/*         biba/10
```

Note: The above file system layout may be different depending on your environment so take care if you are applying this to jail or customized architecture; however, it must be run on every single file system. The `/etc/mac.conf` file requires the following modifications in the main section:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

Enable Networking

It's important to enable networking to enforce MAC to handle incoming/outgoing data and defend the kernel from external access. Implementing this feature requires you to add the following line to `/boot/loader.conf`:

```
security.mac.biba.trust_all_interfaces=1
```

And the following to the network card configuration stored in `rc.conf`. If the primary Internet configuration is done via `DHCP`, this may need to be configured manually after every system boot:

```
maclabel biba/equal
```

Testing the Configuration

- Ensure that the web server and Nagios will not be started on system initialization, and reboot.
- Ensure the `root` user cannot access any of the files in the Nagios configuration directory. If `root` can issue an ls command on `/var/spool/nagios`, then something is wrong. Otherwise a "permission denied" error should be returned.
- If all seems well, Nagios, Apache, and Sendmail can now be started in a way fitting of the security policy. The commands shown on Figure 18 will make this happen.
- Check the log files or error messages to make sure everything is fine. Use the sysctl utility to disable the mac_biba security policy module enforcement and try starting everything again, like normal.

Conclusion

The MAC Security Framework is a vital security enhancement in OS cryptology. It introduces run-time policy en-

References

- FreeBSD Handbook *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/*
- TrustedBSD MAC Project *http://www.trustedbsd.org/mac.html*
- Mandatory Access Control *http://www.freebsd.org/doc/en/books/handbook/mac.html*
- Robert Watson, Brian Feldman, Adam Migus, Chris Vance, "Design and Implementation of the TrustedBSD MAC Framework," Conference and Exposition (DISCEXIII), Washington, DC, 2003. Proceedings published by IEEE.
- Robert Watson, Wayne Morrison, Chris Vance, and Brian Feldman, "The TrustedBSD MAC Framework: Extensible Kernel Access Control for FreeBSD 5.0," USENIX Annual Technical Conference (FREENIX'03).
- FreeBSD Handbook *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/*
- *http://docs.freebsd.org/doc/6.2-RELEASE/usr/share/doc/pl_PL.ISO8859-2/books/handbook/mac-implementing.html*
- *www7.tw.freebsd.org/doc/handbook/mac.html*

forcement techniques. However, MAC security does not handle multi-threading security on behalf of one user. This new requirement was an incentive to start UNIX Capsicum.

Related Work

One of the most common projects running for securing Linux systems is Security Enhanced Linux, which is a set of patches to the Linux kernel and some utilities to incorporate a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel.

MOHAMED FARAG

Mohamed Farag is Technical Consultant at Summa Technologies. Before joining Summa, Mohamed was a programmer analyst at FedEx Services for 10 months. In 2012, Mohamed became a member of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), International Association of Computer Science and Information Technology (IACSIT), Computer Science Teachers Association (CSTA ACM) and Academy & Industry Research Collaboration Center (AIRCC). By the end of 2012, Mohamed was selected to be a reviewer for the International Journal of Computer Science and Information Technology (IJCSIT) and program committee at AIRCC. Mohamed has been an active contributor in the FreeBSD community since May 2010 and has led the ArabBSD project since June 2011. Awards that Mohamed has won include Google Summer of Code and Best Programming Project in Renewable Energy Conference in Egypt.

Using Nginx and Naxsi

for Security and High Availability

Nginx is a lightweight and powerful HTTP proxy, mail proxy, and reverse proxy server for UNIX-like systems that can also be used as an excellent High Availability (HA) and cluster system. Naxsi is an Nginx Web Application Firewall (WAF) created to mitigate web application vulnerabilities by using an intelligent resource to generate whitelist rules.

What you will learn…

- Basic setup of the Nginx cluster/load balancer
- How the Naxsi web application firewall works

What you should know…

- Basic FreeBSD shell command line

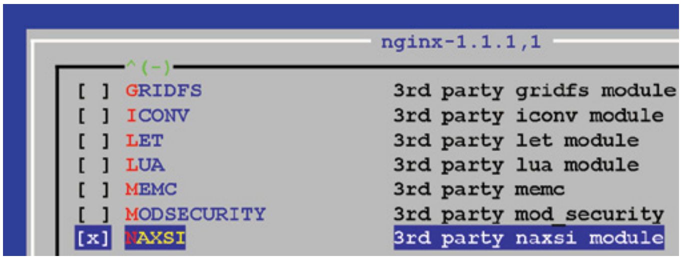
This article will cover basic Nginx HA configuration and show how to improve web security by using the Naxsi *Web Application Firewall* (WAF). Nginx, written by Igor Sysoev, is a forward and reverse HTTP proxy server that can also work as a mail proxy. You can learn more about Nginx and its features at the Nginx wiki page: *http://wiki.nginx.org.*

Before the applications are installed, it is necessary to update the FreeBSD ports tree to use the latest Nginx version. Update Ports:

```
#cd /usr/ports
#portsnap fetch update
```

Install Nginx and Naxsi:

```
#cd /usr/ports/www/nginx
#make install clean
```



```
                                  nginx-1.1.1,1
       ^(-)
   [ ] GRIDFS                    3rd party gridfs module
   [ ] ICONV                     3rd party iconv module
   [ ] LET                       3rd party let module
   [ ] LUA                       3rd party lua module
   [ ] MEMC                      3rd party memc
   [ ] MODSECURITY               3rd party mod_security
   [x] NAXSI                     3rd party naxsi module
```

Figure 1. *Installing Nginx with Naxsi support*

Clustering

Enabling the cluster/load balancer feature is as simple as including the *upstream* directive on the *vhost* information. Listing 1 shows a cluster configuration.

The *acme* entity will be referenced by the *proxy_pass* parameter (Listing 2), and the hosts included in the upstream directive will respond to the requests sequentially.

Listing 1. *Configuring the cluster*

```
upstream acme {
        server 192.168.0.2;
        server 192.168.0.3;
        server 192.168.0.4;
        }
```

Listing 2. *The acme entity being used in the proxy_pass parameter*

```
server {
  listen acme:80;
  access_log /var/log/nginx/nginx.log;

  location / {
    proxy_pass http://acme;
  }
}
```

We can preserve the connection in the same backend using the *ip_hash* parameter (Listing 3) which, for example, is important for authenticated sessions.

Priority and failover

The parameter weight is responsible for defining the backend priority. In Listing 4, the first three requests will be sent to the server 192.168.0.2, the fourth and fifth to 192.168.0.3, and the sixth request to 192.168.0.4.

The failover is enabled using the parameters `max_fails` and `fail_timeout`. `Max_fails` defines the total request failures allowed within the `fail_timeout` time interval. After that, requests are sent to the next backend. To take servers offline from the cluster, use the parameter `down` (Figure 5).

So far we have configured a basic cluster/load-balancing system using Ngnix and now, we will include a security layer to protect the web applications against common web attacks.

Naxsi

Naxsi is a *Web Application Firewall* (WAF) for Nginx created by Thibault Koechlin. A WAF is a tool that listens to the HTTP conversations and applies a set of rules that protect web applications against common attacks such as SQL injection, cross-site scripting, Directory Traversal and Remote File Inclusion. Other examples of WAFs are:

- Modsecurity
- Ironbee

A great advantage of Naxsi is that it follows the positive security model, learning how the applications work and creating rules based on behavior (whitelist-based). It does not use attack signatures like antivirus software does. Naxsi was automatically installed during the compilation of Nginx.

In Listing 6, we created the basic structure to enable Naxsi. In Listing 7, the learning mode will analyze the

Listing 3. *Using the parameter ip_hash to preserve the client request*

```
upstream acme {
        ip_hash;
        server 192.168.0.2;
        server 192.168.0.3;
        server 192.168.0.4;
}
```

Listing 4. *Using the parameter weight to define the server priority*

```
upstream acme {
        ip_hash;
        server 192.168.0.2 weight=3;
        server 192.168.0.3 weight=2;
        server 192.168.0.4;
}
```

Listing 5. *Defining the failover parameters and disabling an offline server using the parameter down*

```
upstream acme {
        ip_hash;
        server 192.168.0.2 max_fails=3  fail_
                    timeout=30s;
        server 192.168.0.3;
        server 192.168.0.4 down;
}
```

Listing 6. *Creating the basic structure. The CheckRule directive specifies that any request having an attack score greater than or equal to 8 will be denied*

```
# mkdir /var/tmp/nginx/
# vi /usr/local/etc/nginx/naxsi.rules
LearningMode; #Enables learning mode
SecRulesEnabled;
#SecRulesDisabled;
DeniedUrl "/RequestDenied";
## check rules
CheckRule "$SQL >= 8" BLOCK;
CheckRule "$RFI >= 8" BLOCK;
CheckRule "$TRAVERSAL >= 4" BLOCK;
CheckRule "$EVADE >= 4" BLOCK;
CheckRule "$XSS >= 8" BLOCK;
```

Listing 7. *Enabling the Naxsi WAF*

```
include /usr/local/etc/nginx/naxsi_core.rules;
      ...
location / {
     include    /usr/local/etc/nginx/naxsi.rules;
     proxy_pass  http://acme/;
     ...
     }
#Naxsi Learning Mode
  location /RequestDenied {
   return 500;
   }
```

Listing 8. *Installing nx_util dependencies*

```
# cd /usr/ports/lang/python
# make install clean
# cd /usr/ports/databases/sqlite3
# make install clean
```

behavior of the application to create new rules. The `nx_util` is another useful tool that parses the web site error logs, updates a SQLite database, generates new whitelist rules, and exports the data to the screen or an HTML file (Listing 8-10 and Figure 2).

```
tail /var/log/nginx-error.log | ./nx_util.py -c /usr/local/
etc/nginx/nx_util-0.3/nx_util.conf -l -i -o -H acme.html
```

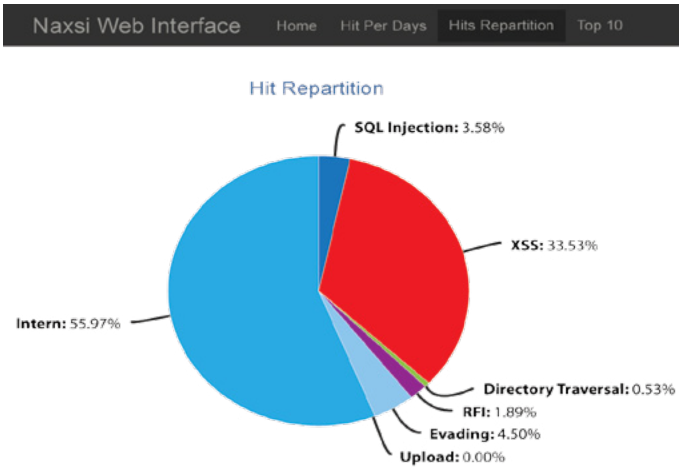


Figure 2. *HTML page generated using nx_util*

On The Web

- Nginx official Web Site: *http://nginx.org/*
- Nginx Wiki page: *http://wiki.nginx.org/*
- Naxsi Web site: *https://code.google.com/p/naxsi/*

An example of *nginx.conf* can be downloaded here: *http://alexos.org/files/nginx.conf*.

Conclusion

Now we have a complete cluster/load balancing system, and we have secured our web applications using Nginx and Naxsi WAF. Remember to create file and database replication for a completely fault-tolerant system.

As this is my first article in BSD Magazine, many thanks to all for reading it!

ALEXANDRO SILVA AKA ALEXOS

Alexandro Silva aka Alexos lives in Salvador, Bahia, Brasil. He is an Information Security Consultant at iBliss Segurança & Inteligência. He has been using FreeBSD since the 4.11 release and can be reached online at http://alexos.org.

Listing 9. *Configuring nx_util*

```
# cd /usr/local/etc/nginx/nx_util-0.3
# vi nx_util.conf
[nx_util]
data_dir=/usr/local/local/etc/nginx/nx_util-0.3/nx_datas
database_dir=
naxsi_core_rules=/usr/local/etc/nginx/naxsi_core.rules
# cd /usr/ports/databases/sqlite3
# make install clean
```

Listing 10. *After executing a web vulnerability scan on the ACME web site, nx_util generates new whitelist rules*

```
#  tail /var/log/nginx-error.log | ./nx_util.py -c /usr/
local/etc/nginx/nx_util-0.3/nx_util.conf -l -i -o
Using stdin.
Committing to db ...
######### Optimized Rules Suggestion #########
# total_count:7 (7.87%), peer_count:1 (100.0%) | parenthesis,
                      probable sql/xss
BasicRule wl:1011 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:username”;
# total_count:7 (7.87%), peer_count:1 (100.0%) | parenthesis,
                      probable sql/xss
BasicRule wl:1011 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:7 (7.87%), peer_count:1 (100.0%) | parenthesis,
                      probable sql/xss
BasicRule wl:1010 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:6 (6.74%), peer_count:1 (100.0%) | obvious probe
BasicRule wl:1202 “mz:$URL:/vulnerabilities/brute/|$ARGS_VAR:login”;
...
BasicRule wl:1205 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:1 (1.12%), peer_count:1 (100.0%) | http:// scheme
BasicRule wl:1100 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:1 (1.12%), peer_count:1 (100.0%) | mysql comment
                      (--)
BasicRule wl:1007 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:1 (1.12%), peer_count:1 (100.0%) | mysql keyword (|)
BasicRule wl:1005 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
# total_count:1 (1.12%), peer_count:1 (100.0%) | uncommon hex
                      encoding (%00 etc.)
BasicRule wl:10 “mz:$URL:/vulnerabilities/brute/|$ARGS_
                      VAR:login”;
```

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES
1.925.240.6652

$39.95
FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

$29.95
PC-BSD 9.1 DVD

$49.95
Save a BUNDLE!
The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD

$99.95
The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD

Stylish Dress Attire
Look Your Professional Best

Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD $39.95

CD Set Contains:

Disc 1	Installation Boot LiveCD (i386)
Disc 2	Essential Packages Xorg (i386)
Disc 3	Essential Packages, GNOME2 (i386)
Disc 4	Essential Packages (i386)

FreeBSD 9.0 CD	$39.95
FreeBSD 9.0 DVD	$39.95

FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1	$29.95
FreeBSD Subscription, start with DVD 9.1	$29.95
FreeBSD Subscription, start with CD 9.0	$29.95
FreeBSD Subscription, start with DVD 9.0	$29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD	$29.95
PC-BSD Subscription	$19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide)	$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)	$39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes)	$59.95
The FreeBSD Handbook, Both Volumes & FreeBSD 9.1	$79.95

PC-BSD 9.0 Users Handbook	$24.95
BSD Magazine	$11.99
The FreeBSD Toolkit DVD	$39.95
FreeBSD Mousepad	$10.00
FreeBSD & PCBSD Caps	$20.00
BSD Daemon Horns	$2.00

Bundle Specials!
Save $$$

Just Plain Fun
Mousepads & Novelty Horns

BSD Magazine
Available Monthly

FreeBSD Mall

For even MORE items
visit our website today!
www.FreeBSDMall.com

FreeBSD Programming Primer – Part 6

In the sixth part of our series on programming, we will design a basic menu navigation system and style it with CSS.

What you will learn…

- How to configure a development environment and write HTML, CSS, PHP and SQL code

What you should know…

- BSD and general PC administration skills

So far in this series, we have focused on adding and displaying standard HTML pages which have been pulled from our database. We are now going to shift directions and start to look at the user interface of the CMS itself. Traditionally, menu links were hard coded into pages, which not only made long-term maintenance time-consuming but also error-prone. By leveraging the power of a database back end, we can easily extract the title and section of pages we want to display and if desired, include or exclude that content from the menu. For flexibility, we will also include the facility to add disparate links to other sites, etc.

Many sites now use multi-level menus which are driven by a combination of SQL, Javascript / Jquery and CSS. Later on in the series, we will look at using Jquery to add this functionality, but for now we will concentrate on a block navigation menu that is displayed alongside the main content.

The SQL

To demonstrate, let's spin up a MySQL session and take a look at our content. At the shell prompt, login to MySQL and run some queries (Listing 1 – 2).

By using the UNION keyword, we can combine the output of both SELECT statements into one result. This would be fine if we had a small site with not much content, but as the site grows, the menu would become unmanageable in size. We could build the interface with a drop-down and filter by section, but we would just be postponing the inevitable. An additional improvement would be to use a combination of a content type filter and a pager with the MySQL LIMIT keyword, restricting the display to a certain number of items. This would help in the final design

Listing 1. *Logging in to MySQL*

```
#dev mysql -u bsduser -pcmsdbpassword
```

Listing 2. *Selecting our content*

```
mysql> use freebsdcms;
mysql> (SELECT 'news' AS contenttype, id, title FROM
                   news) UNION (SELECT 'pages' AS
contenttype, id, title FROM pages);

+-------------+----+----------------------+
| contenttype | id | title                |
+-------------+----+----------------------+
| news        |  1 | My first page       |
| news        |  2 | My second page       |
| news        |  3 | Article 5 - Using CSS |
| pages       |  1 | My first page       |
| pages       |  2 | My second page       |
+-------------+----+----------------------+
5 rows in set (0.00 sec)
```

and theming of the site, as we will know exactly how much browser real estate would be occupied by the menu itself even if the content expanded rapidly.

The remaining issues are how to add disparate links and whether we want to display the content in the menu at all. For example, we might have an error page that only is displayed when the content is not found. While it would be useful to store this in the database, displaying it in the menu would be rather pointless. The question is where to store this data? We could have a separate menu table, with the ID of each page and a numeric flag (0, 1) to represent do not display in the navigation menu or include in the menu. We would then have to maintain 2 tables when content is added and removed. This could be easily accomplished using MySQL triggers. Alternatively, we could store the page status in the relevant content tables (e.g. news, pages) with a flag (0,1,2) to represent “do not publish”, “publish but do not show in menu”, and “publish and

Listing 3. *Creating FAQ’s table and adding status flag*

```
mysql> CREATE TABLE faqs LIKE news;
mysql> ALTER TABLE faqs ADD status INT DEFAULT 0 AFTER
                   content;
```

Listing 4. *Adding auto increment to the FAQ table*

```
mysql> ALTER TABLE faqs CHANGE id id INT(11) AUTO_INCREMENT;
```

Listing 5. *Adding data to the FAQ table*

```
mysql> INSERT INTO faqs(id, title, heading, content,
                   status, timestamp) VALUES(‘’,
‘FAQ 1’, ‘First FAQ’, ‘Aenean volutpat, ligula vitae
                   laoreet dapibus’,2,’’);
```

Listing 6. *Amending the remaining tables*

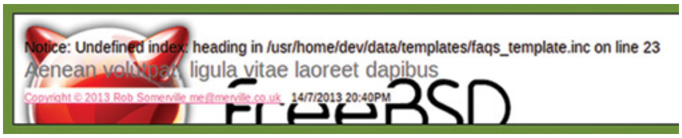
```
mysql> ALTER TABLE pages ADD status INT DEFAULT 0 AFTER content;
mysql> ALTER TABLE news ADD status INT DEFAULT 0 AFTER content;
mysql> ALTER TABLE pages CHANGE id id INT(11) AUTO_INCREMENT;
mysql> ALTER TABLE news CHANGE id id INT(11) AUTO_INCREMENT;
```

Listing 7. *Our 3 table content*

```
mysql> (SELECT ‘news’ AS contenttype, id, status, title
                   FROM news) UNION (SELECT
‘pages’ AS  contenttype, id, status, title FROM pages)
                   UNION (SELECT ‘faqs’ AS
contenttype, id, status, title FROM faqs);

+-------------+----+--------+-----------------------+
| contenttype | id | status | title                 |
+-------------+----+--------+-----------------------+
| news        |  1 |      0 | My first page         |
| news        |  2 |      0 | My second page        |
| news        |  3 |      0 | Article 5 - Using CSS |
| pages       |  1 |      0 | My first page         |
| pages       |  2 |      0 | My second page        |
| faqs        |  1 |      2 | FAQ 1                 |
| faqs        |  2 |      0 | FAQ 2                 |
| faqs        |  3 |      1 | FAQ 3                 |
| faqs        |  4 |      2 | FAQ 4                 |
| faqs        |  5 |      2 | FAQ 5                 |
| faqs        |  6 |      2 | FAQ 6                 |
| faqs        |  7 |      2 | FAQ 7                 |
| faqs        |  8 |      2 | FAQ 8                 |
| faqs        |  9 |      2 | FAQ 9                 |
| faqs        | 10 |      2 | FAQ 10                |
+-------------+----+--------+-----------------------+
15 rows in set (0.00 sec)
```

Listing 8. *Updating the news and pages status*

```
mysql> UPDATE news SET status = 1;
mysql> UPDATE pages SET status = 2;
mysql> (SELECT ‘news’ AS contenttype, id, status, title
                   FROM news) UNION (SELECT
‘pages’ AS  contenttype, id, status, title FROM pages)
                   UNION (SELECT ‘faqs’ AS
contenttype, id, status, title FROM faqs);

+-------------+----+--------+-----------------------+
| contenttype | id | status | title                 |
+-------------+----+--------+-----------------------+
| news        |  1 |      1 | My first page         |
| news        |  2 |      1 | My second page        |
| news        |  3 |      1 | Article 5 - Using CSS |
| pages       |  1 |      2 | My first page         |
| pages       |  2 |      2 | My second page        |
| faqs        |  1 |      2 | FAQ 1                 |
| faqs        |  2 |      0 | FAQ 2                 |
| faqs        |  3 |      1 | FAQ 3                 |
| faqs        |  4 |      2 | FAQ 4                 |
| faqs        |  5 |      2 | FAQ 5                 |
| faqs        |  6 |      2 | FAQ 6                 |
| faqs        |  7 |      2 | FAQ 7                 |
| faqs        |  8 |      2 | FAQ 8                 |
| faqs        |  9 |      2 | FAQ 9                 |
| faqs        | 10 |      2 | FAQ 10                |
+-------------+----+--------+-----------------------+
15 rows in set (0.00 sec)
```



Notice: Undefined index: heading in /usr/home/dev/data/templates/faqs_template.inc on line 23

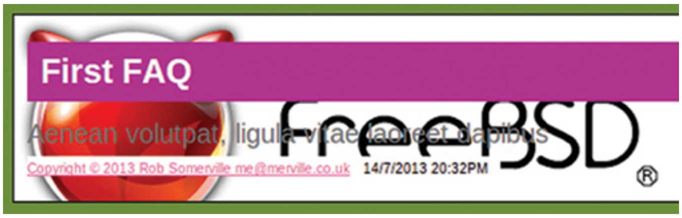
Aenean volutpat, ligula vitae laoreet dapibus

Copyright © 2013 Rob Somerville me@merville.co.uk 14/7/2013 20:40PM

Figure 1. *Bug in core.inc*

show in menu”. Both designs have their good and bad points from the implementation and data integrity viewpoint, but for the sake of simplicity, I will use the latter for our navigation menu.

In the meantime, we have an FAQ definition in our file content.inc but we do not have any table data for it. We will now manually create the table and add 10 random FAQ entries (Listing 3-5). This will result in a new FAQ table with our status field. However, the ID field is not set to auto increment, so we need to change this (Listing 4). Now insert the data (10 entries) – replacing the title, heading and status (0, 1 or 2) as appropriate. We need to repeat the structural amendments for our news and pages tables as well (Listing 6). Let’s check what data we now have in the three tables (Listing 7). As we can see, the news and pages will not be published or displayed in the menu. Change this so the news items are not in the menu but published, but the pages are (Listing 8). Let us check in a browser if FAQ 1, 2 and 3 are displayed. Visit http://yourserverip/faq/1 and you should get an error message “No template”. To rectify this, create a `faqs_template.inc` file in `/usr/home/dev/data/templates` with the following content (Listing 9).



First FAQ

Aenean volutpat, ligula vitae laoreet dapibus

Copyright © 2013 Rob Somerville me@merville.co.uk 14/7/2013 20:32PM

Figure 2. *CSS requires fix for FAQ content type*

Bug alert! If you visit http://yourserverip/faq/1 you will find the page is not rendering correctly (Figure 1). You will receive an error message: Notice: `Undefined index: heading in /usr/home/dev/data/templates/faqs_template.inc on line 23`. If you want to try and diagnose the problem, have a look at core.inc and skip the next code listing. The problem lies in the following code snippet. To fix it, change as follows (Listing 10-11).

If you visit http://yourserverip/faq/1, you will find the page is still not rendering correctly (Figure 2). The reason for this is that the the global CSS doesn’t know about our FAQ content type yet, so we need to modify global.css as follows (Listing 12). You may have to refresh or clear your browser cache to pick this up. This should result in the

Listing 9. *FAQ template*

```
<?php
/*
 *
 * faqs_template.inc
 * Template for our faq content type
 *
 * For content type foo the corresponding template would be:
 * foo_template.inc
 *
 * To display a field:
 * render($theme['name_of_field_as_defined_in_db']);
 *
 * To hide a field omit it from here
 * To change the rendering order, just re-order the fields
 *
 * NOTE: Any content generated by javascript will not
                be managed here
 *      A closing ?> tag is mandatory
 *
 */
render($theme['heading']);
render($theme['content']);
?>
```

Listing 10. *Bad code!*

```
if($pos > $offset){

  $theme[$key] = div($result[$key], $key.'-'.$id, $key);

}
```

Listing 11. *Good code*

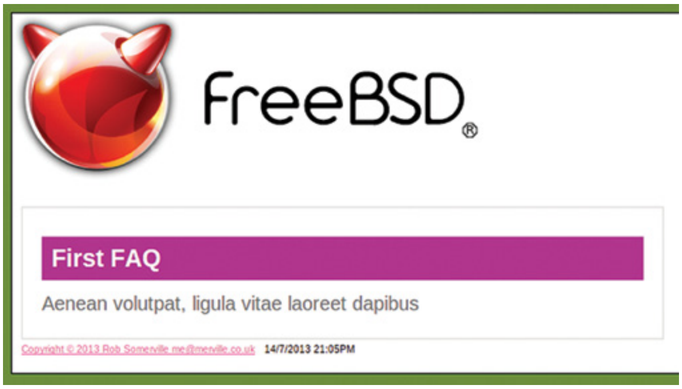
```
if($pos >= $offset){

  $theme[$key] = div($result[$key], $key.'-'.$id, $key);

}
```

Listing 12. *CSS to include FAQ content type*

```
#news, #page, #faq {
```

Listing 13. *Prevent non-published content showing*

```
$sql = "SELECT * FROM $content_type WHERE id='$id' AND
                   status > 0 LIMIT
1";
```



First FAQ

Aenean volutpat, ligula vitae laoreet dapibus

Copyright © 2013 Rob Somerville me@merville.co.uk 14/7/2013 21:05PM

Figure 3. *FAQ working*

correctly rendered content in (Figure 3). However, if we visit http://yourserverip/faq/2 , we will see an FAQ page even though the status is 0. Modify core.inc as follows to fix this (Listing 13). This should now give a “No data” message. If you are still experiencing problems, ensure that the content.inc file is as follows (Listing 14).

Building our menu

How can we remember the filter value selected for the content type? As HTTP is stateless, we could pass the parameter to each page. This would get complex very quickly with multiple menus. A better solution would be to write a cookie to the visitors browser when the content type is filtered. To do this we will use Javascript, and specifically a suite of Jquery libraries. Download jquery-1.10.2.min.js and jquery.cookie.js from the Jquery website. Place these files in the Javascript folder, then modify our source code as follows (Listing 15-18). When you visit http://youripaddress/faq/1, you should see a page similar to Figure 4. Clicking on the FAQ, News or Page button will raise a Javascript dialogue box.

Useful links

- Jquery library: *http://code.jquery.com/jquery-1.10.2.min.js*
- Jquery cookie: *https://github.com/carhartl/jquery-cookie/blob/master/jquery.cookie.js*

In the next part

We will tie the onclick event to writing a local cookie, and extracting the links for the MySQL table. We will also look at using the Jquery library to build a multi-part menu.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

Listing 14. *content.inc*

```
<?php
/*
 *
 * content.inc
 * Defines content types for our CMS
 *
 */

// Define the content type. This must match any tables
                  defined in our
// CMS

$content_types[] = ‘page’;
$content_types[] = ‘faq’;
$content_types[] = ‘news’;

// Map each content type to a table. Each content type
                  must be matched
// to a corresponding table

$content_tables[‘page’] = ‘pages’;
$content_tables[‘faq’] = ‘faqs’;
$content_tables[‘news’] = ‘news’;
```

Listing 15. *header.inc include Jquery support*

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Strict//EN”
“http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd”>
<html xmlns=”http://www.w3.org/1999/xhtml” xml:lang=”en”>
<head>
<meta http-equiv=”Content-type” content=”text/html;
charset=’iso-8859-1’” />
<link rel=”stylesheet” type=”text/css”
href=”/stylesheets/reset.css” />
<link rel=”stylesheet” type=”text/css”
href=”/stylesheets/global.css” />
<script src=”/javascript/jquery-1.10.2.min.js”
type=”text/javascript”></script>
<script src=”/javascript/jquery.cookie.js”
type=”text/javascript”></script>
<script src=”/javascript/preload.js” type=”text/javascript”>
</script>
```

Listing 16. *core.inc*

```
function optimize_callback($buffer) {

    // replace all spaces between tags

    if (OPTIMIZE) {

        $b = preg_replace('~>\s+<~', '><', $buffer);
        $b = preg_replace('/\r\n|\r|\n/', '', $b);
        $b = preg_replace('!\s+!', ' ', $b);

        return $b;

    } else {

        // BUGFIX - Edition 6

        return $buffer;
    }
}
```

Listing 17. *index.php – add include menu.inc*

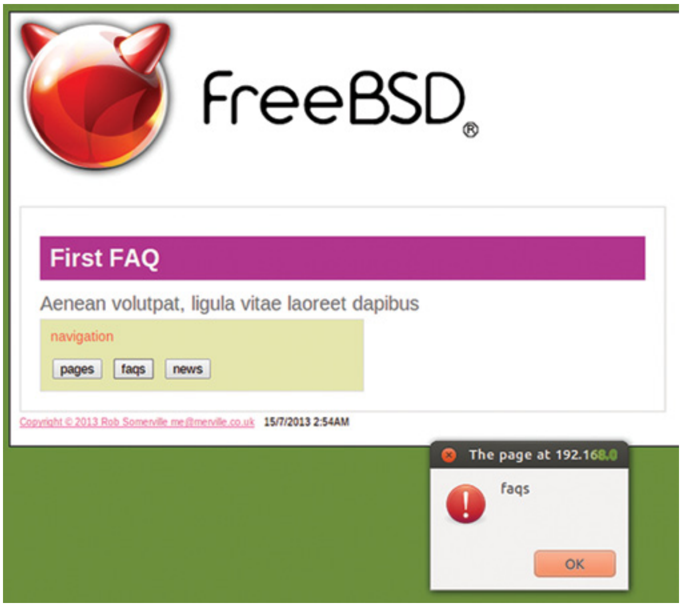
```
// Menu functions
require_once INCLUDES.'menu.inc';
```

Listing 18. *menu.inc*

```
<?php

function menu($type) {

    require INCLUDES . 'content.inc';

    if ($type == 'navigation') {

        // Build select statement for each content type
                   in turn
        // Omit the UNION keyword on the last item

        $offset = 1;
        $categories = count($content_tables);
        $sql = '';
        $option = '';

        foreach ($content_tables as $contenttype) {

            // Build the option for the content type

            $option .= '<button onclick="window.alert(\''.
          $contenttype.'\')">'.$contenttype.'</button> &nbsp;';

            $offset ++;

        }

        $menu = '';

        $menu .= '<div class ="menu-' . $type . '">';
        $menu .= '<h2>' . $type . '<h2>';
        $menu .= '<p>&nbsp;</p>';
        $menu .= $option;
        $menu .= '</div>';

        return $menu;
    }
}
```

Listing 19. *Menu CSS add to global.css*

```
.menu-navigation {
    border: 1px solid #DADADA;
    padding: 10px;
    width: 50%;
    background-color: #E5E6AD;
}

h2 {
    color: tomato;
    font-weight: 600;

}
```



First FAQ

Aenean volutpat, ligula vitae laoreet dapibus

navigation

Copyright © 2013 Rob Somerville me@merville.co.uk 15/7/2013 2:54AM

The page at 192.168.0

faqs

Figure 4. *FAQ with Javascript onclick buttons*

VERISIGN®

vBSDCon · SAVE THE DATE!

DULLES, VA · OCTOBER 25-27, 2013

Please join us October 25-27, 2013 at the Hyatt in Dulles, Virginia for the first biennial vBSDCon event. This exciting weekend will bring together members of the BSD community for a series of roundtable discussions, educational sessions, best practice conversations, and exclusive networking opportunities. See below for details on this industry weekend not to be missed:

AGENDA

- **Friday, October 25:** Evening Reception
- **Saturday, October 26:** General Session, Birds of a Feather Sessions
- **Sunday, October 27:** General Session, Breakout Sessions

WHO SHOULD ATTEND

- Developers
- Engineers
- Administrators
- Innovators

TOPICS

- PkgNG w/ Baptiste Daroussin
- A comprehensive look at bsdinstall with Devin Teske
- Netflix Demo/Presentation with Scott Long
- netmap with Luigi Rizzo
- Migration from GCC to LLVM/Clang with David Chisnall

REGISTRATION INFORMATION WILL BE SENT TO YOU IN MAY!

Questions? Please contact: eventsteam@verisign.com

VerisignInc.com

Elliptic Curve

Cryptography and Basic Applications in BSD OS

Over the past century, new IT discoveries and top technologies have been incorporated into our daily lives. As a result, one must decide whether computer security is essential or superfluous. In these times, when the Internet allows users to communicate easily around the world, many people use it to trade in goods or to protect against intruders. That's why cryptological techniques and algorithms are an essential need.

What you will learn…

- Basic terms of Public Key Cryptography.
- Elliptic curves and their application to cryptographic methods.
- Foundations of Elliptic Curve Cryptography.
- Basic application of ECC public key techniques.
- Development of ECC cipher and digital signature systems for NetBSD OS.

What you should know…

- Basic knowledge of public key cryptographic methods.
- Basic knowledge of Abstract Mathematics, Modular Arithmetic, Algebraic Structures, and Group/Field Theories.
- User-level background of NetBSD OS (a Unix-like OS).

Symmetric Key Cryptography (SKC) techniques perform better than the traditional *Public Key Cryptography* (PKC) algorithms such as RSA. For this reason, introducing a new PKC approach based on elliptic curves could become important in the near future because these methods provide better performance over the former ones, although they are not as good as SKC techniques.

Although all these aspects are known by the IT users community, the most modern cryptographic techniques, such as those based on public-key algorithms like RSA, DSA or IDEA, show great promise to harden existing systems. In this context, public-key algorithms based on *Elliptic Curves Theory* (ECT) will be the main focus of this article.

Hence, the main purpose of this paper is not only to present the associated terms and definitions of the *Elliptic Curves Cryptography* (ECC) methods, but also to demonstrate that these techniques are a subset of the general public-key cryptographic methods. Due to this fact, it's simple to deploy and use in Unix-like systems and, in particular, within NetBSD.

Just allow me a digression, as I would like to apologize to the reader for the approach followed, which is sometimes excessively abstract. A background knowledge of the science behind the ECC methods will help you easily understand the topics introduced here.

Cryptography

Cryptography is the scientific discipline of ensuring the secret transmission of information so that it can be read only by the message sender and receiver. There is a need, as old as human history, to protect personal information and privacy. The first reference to this topic can be dated back to 1900 BC in Egypt. Currently, cryptography must meet four basic requirements:

- Privacy: Nobody else apart from the sender and receiver of a message can access the content of the message.
- Non-repudiation: This mechanism ensures the identity of the sender.
- Authentication: Each of the participants involved in a communication process must prove their identity.
- Integrity of information: Ensures that the content has not been modified during transmission.

Cryptographic methods must not only protect the data from third parties but must also guarantee the integrity of data transmitted, ensure the identities of both sender and receiver and allow both of them to communicate. There are three types of cryptographic schemes that fulfill these four requirements. According to the well-known terms used in the literature, we shall refer to non-ciphered data as cleartext, and call encrypted data ciphertext. For third parties, it is common to use terms such as eavesdropper or man-in-the-middle.

Cryptographical Algorithms and Types

As explained in the previous paragraphs, the three types of cryptographic methods exist for two main purposes: to exchange information between two entities so that nobody else can read it and to verify that the information or data has not been modified during delivery. Within the first category, we have Symmetric Key Cryptography/Public Key Cryptography techniques and for the second one, we count on hash-functions. Let us briefly describe these three cryptological techniques.

Symmetric-Key Cryptography

For *Symmetric-Key Cryptography* (SKC), both sender and receiver share the same key to cipher and decipher (Figure 1).

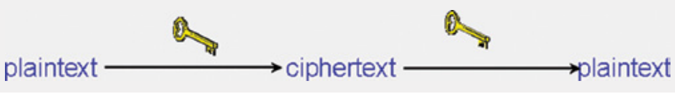


Figure 1. *Symmetric Key Cryptography (SKC)*

That means SKC involves two people using the same (private) key for both encrypting and decrypting information. Because SKC uses a single key, algorithms based on this approach are faster than PKC and are easier to implement. However, the main weakness of SKC relies on the fact that before starting secure communication, both parties need to have a copy of the shared key. Under some circumstances, that can be impossible.

Public-Key Cryptography

Public-Key Cryptography (PKC), in which a pair of different keys are used. These keys are designated as public and private, depending on their usage (Figure 2).

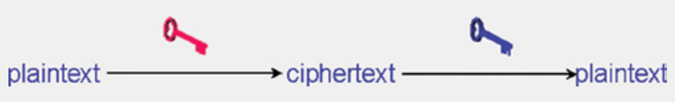


Figure 2. *Public Key Cryptography (PKC)*

In contrast to the SKC approach, using Public Key Cryptography (PKC) algorithms does not require that the public key remain secure, and it does not matter at all if anybody other than the two parties in the secure communication process knows it. In fact, it is possible to go further by using both PKC and SKC for secure communications. In this way, two people create a secure channel by using PKC and then they can share a key so they can use SKC for further communication. This mixed approach takes advantage of the best of both worlds: performance and security.

Hash Functions

Hash functions (Digests), also known as one-way functions, map a value to a message. In this schema, the concept of key makes no sense as the purpose of these functions is not to recover the message but to ensure that the contents of the message have not been altered.

These functions must exhibit a basic property: any modifications in the message produce high variations in the final numeric result (Figure 3).

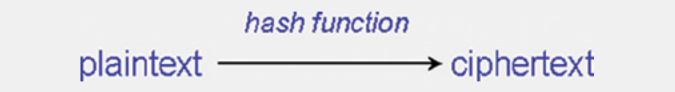


Figure 3. *Hash Functions (Digest functions)*

Categories of Cryptographic Algorithms

The reason that three different categories of cryptographic algorithms have been developed is motivated by their final applications. Hence, while hash functions have been created to ensure referential integrity of data, SKC was conceived for message ciphering and PKC plays an important role in user authentication by means of the users' keys exchange.

Ultimately, SKC shows a higher performance than PKC algorithms as the latter takes three times longer to execute in comparison with SKC. As we explained in the introduction, this will be the motivation to look for better performance PKC algorithms.

As explained in the previous section, SKC algorithms like DES and its variants 3DES, use a unique key to encrypt and decrypt a message. Although this approach is easier to implement, there's a handicap since before starting secure communications between two partners, both of them must have a copy of this key. Moreover, this key must be kept secret between the two participants so that a third party can't access the contents of the message.

Although the use of SKC algorithms implies better performance, the price to pay is the inherent risk when two parties must share the same key and are usually not in touch.

PKC algorithms are based upon the existence of a pair of complementary keys, named public and private key: the use of one of them is undone by using the other one.

The difference between the public and the private key is merely a matter of usage, as both keys are implemented as nothing more than integers from a mathematical point of view. Computers represent these integer numbers as strings.

In contrast to SKC algorithms, PKC algorithms allow us to bypass the requirement of SKC that two actors must share the same key. The tradeoff in this case is the performance we would get by using SKC. Hence it is a matter of compromise to make the right decision about which to use.

Also known as one-way cryptography, Hash functions are based on the existence of the inverse mapping for a map that associates a message/file with an integer number or message digest. These hash mappings have two essential properties for security applications:

- A small modification in the message produces a noticeable change in the final digest.
- The calculus of the inverse of this function or mapping is very difficult, making it almost impossible to recover the original message starting from the digest of it.

These algorithms are widely used to check the integrity of the transmitted data. Some known examples are the *Message Digest* (MD) algorithms such as MD2; MD4, used to encrypt passwords in MS Windows; and MD5, developed to eliminate MD4 weaknesses.

An example of the use for each schema explained above is given in Figure 4.

Up until now, the mathematical tools used to develop new cryptographic techniques were based on Modular Arithmetic and it is precisely this discipline the most important point to introduce a new approach for PKC methods, provided by Algebraic Geometry and materialized in some special functions termed Elliptic Curves. PKC based on elliptic curves is known as *Elliptic Curve Cryptography* (ECC).

Elliptic Curve Cryptography (ECC)

In 1985, Elliptic Curve Cryptography (ECC) was proposed by Victor Miller of IBM and Neal Koblitz of Washington University as an alternative to the public key algorithms based on modular arithmetic and the factoring of big primes as a way of obtaining more robust and less computationally demanding algorithms. Elliptical cryptography is based on the resolution of the problem of discrete logarithm of the curve. However, before going deeply into these concepts, we need to let go of our mathematical prejudices and start thinking abstractly.

Groups Theory and Modular Arithmetic

Keep in mind before reading the following sections that we will not deal with numbers but with abstract entities. A *group* is nothing more than an algebraic structure defined on a set G whose elements are *points*.

Since the elements of a set are points – it is possible to invent operations that associate a couple of points of the above mentioned set with the third one of the same set, which is known in Mathematics as an *internal operation*,

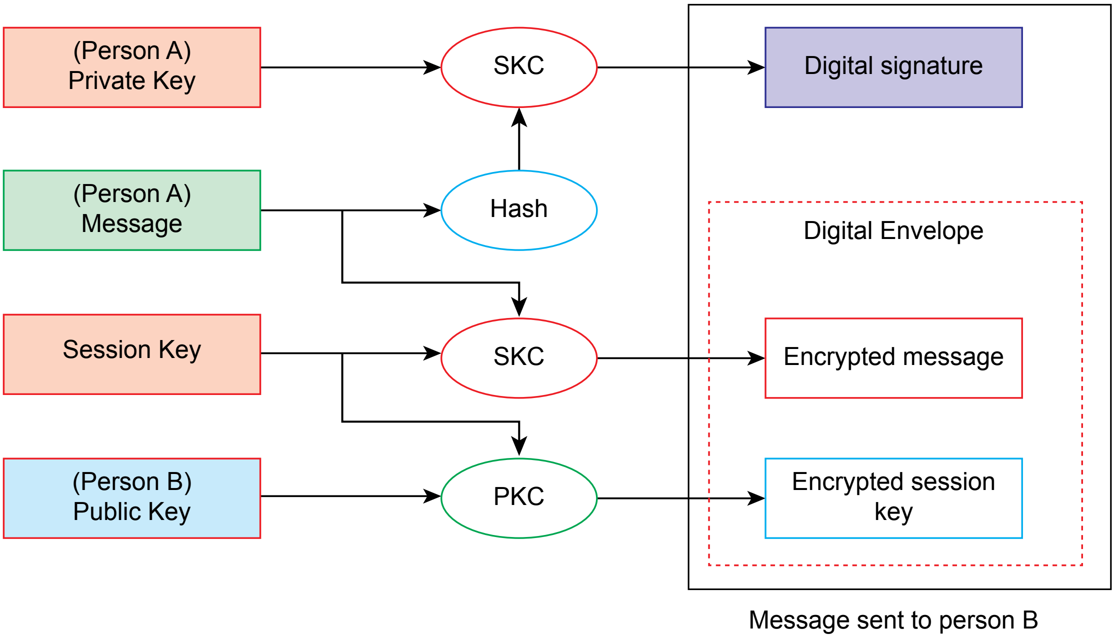


Figure 4. *Application for SKC, PKC schemas and hash (digest) functions*

which is nothing more than one which associates at every pair of points of G another point of the same set endowed with the following properties:

- Associability
- Unit element
- Inverse element

Furthermore, if the commutative property happens, we will say of (G, +) it presents a structure of the abelian group. Let's think about one of these sets because we will use it hereinafter. Let's see the usefulness of another concept that is studied in Elementary Mathematics and to which it does not explain how to get profit of it, as are the relations of equivalence, which allow to establish classes of equivalence in the above mentioned set. Let's take the whole numbers; let's choose a non-negative whole number, m, and establish a relation of equivalence:

```
x R y if and only if (x-y) mod m=0
```

That means two integer numbers are related, if when we divide its difference by m, the remainder is equal to 0. In this way, we could have the following equivalence classes:

```
[0]={0, m, 2m, 3m, 4m}
[1]={1, m+1,2m+1,3m+1, ... }
[m-1]= {m-1, 2(m-1), 3(m-1), ...}
```

That is to say, we have moved from a set Z with infinite elements to another one consistent of the resultant classes of equivalence of introducing the previous relation that besides turns out to be finite. The above mentioned set named `Zn` and its elements are [0], [1]... [m 1]. Let's define now a "sum" operation that acts on two of these classes of equivalence, associating the third one of the following form:

```
[i]+[j]=[k] if (i+j) R k (i.e. (i+j-k) mod m = 0)
```

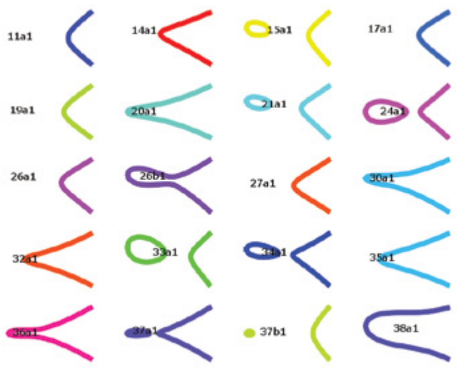


Figure 5. *Elliptic curves with coefficients a, b real*

At first glance, these concepts seem to be somewhat obscure and complex since this operation on the set `Z _ m` is the base of most of the algorithms of public key as Diffie-Hellman, DSA and RSA and is going to be the starting point for the algorithms of public key based on the Theory of Elliptic Curves; algorithms which lead to ECC.

Elliptic curves were originally introduced by the German mathematician Bernhard Riemann in the 19th century and these combine also Number Theory as Algebraic Geometry. From this point of view, an elliptic curve is nothing more than the set of points obeying a 3rd degree polynomial equation:

$$y^2=x^3+ax+b$$

The shape of these curves depends on the choice for the parameters a and b, as can be seen in Figure 5.

These curves have an interesting property: small variations in the parameters a and b produce big variations in the points of the above mentioned curve. Let's define an operation on one of these curves.

We might show which is the natural number n such that the point `Q=nP` and this one is the problem ECDL. The foundation of the algorithms ECC takes root in the fact that a person can know the points P and Q, but determining the number n is intrinsically difficult from a computational point of view.

Replacing the roles from a traditional point of view for IT security, we have two entities acting as our PKC keys:

- Point Q Public Key ECC
- Integer n Private Key ECC

In summary, we have defined all required elements to make up a public-key algorithm based on EC theory which is very different from those ones based on pure modular arithmetic that try to exploit big prime numbers, such as RSA or DSA. These ECC algorithms will provide the same level of security but using smaller key sizes compared to their predecessors, as shown in Table 1.

Table 1. *Key sizes comparison between RSA y ECC algorithms*

Key bits RSA	Decrypt Time (MIPS Years)	Key bits ECC	Key bits rate between RSA / ECC
512	104	106	05:01:00
768	108	132	06:01:00
1,02	1011	160	07:01:00
2,05	1020	210	10:01:00
21,00	1078	600	35:01:00

As ECC key sizes are significantly smaller than those for RSA, the performance of ECC methods is higher, resulting in better response time as well as less memory and network bandwidth consumption. Nevertheless, ECC is much more efficient than RSA for the signature (company) and deciphered, but slower in counterpart for operations of monitoring signature (company) and encrypted; even with everything, for applications in which (MIPS) gives priority to the shortage of resources of memory, bandwidth, or computer power, ECC would be the reasonable algorithm; for example in the case of embedded systems.

Elliptic Curves. A Brief Introduction

We are going to give a quick introduction on one of the most exciting topics of pure mathematics: the algebraic theory of elliptical curves, which are nothing more than algebraic expressions. Let a and b be integer numbers and consider the equation:

$$y^2=x^3+ax+b$$

Let's suppose for a moment, that instead of using real numbers for the above mentioned parameters a and b (as was shown in Figure 5), we assume that the above mentioned parameters [0], [1], [2] belong to a set $Z\_p = \{.... [p-1]\}$, such and since it has been defined in the previous paragraph, with p being a prime number; that is, divisible by itself and one, on that we define an operation $(\bmod\ p)$ of the following form:

$$x \pmod p = \text{remainder of the integer division of } x \text{ by } p$$

Let Z_p be the set of elements $x \pmod p$. Now we can focus our attention to the family of elliptic curves:

$$y^2=x^3+ax+b$$

where a, b belong to the set Z_p already defined and these two parameters meet the following relationship:

$$4a^3 + 27b^2 \mathrel{!}=0 \pmod p$$

Also, we need another special point placed at infinite that we name O. It is feasible to define an internal operation "+" which maps two points P, Q over this curve to a third one R also belonging to it and, what is more, it is amazing that this new operation we have just created verifies these three well-known properties:

- Unit Element: P+O=O+P=P
- Inverse Element: For all point P there exists a point Q such that P+Q=Q+P=O
- Associativity: Given three points P, Q and R, the equation (P+Q)+R=P+(Q+R) holds

A set of points provided with an operation adds that satisfies these three properties: it is said to possess the algebraic structure of group and if besides, it verifies the commutative property; that means, for any couple of points P, Q there is fulfilled that P+Q=Q+P, it is said that the above mentioned group is commutative or abelian. With it we have set all the formal elements for the definition and development of algorithms of cryptography by means of the employment of elliptical curves or ECC.

From Points to Numbers

Let's define a "sum-like" operation on points of an elliptical curve with coefficients defined on Z_m of the following form.

Let P and Q be two points characterized by their coordinates in $Z\_m \times Z\_m$ $P = (x1, y1)$, $Q = (x2, y2)$. The above mentioned operation adds up, associates with the point $R=P+Q$ the coordinates $(x3, y3)$ calculated of the form indicated below:

$$x3 = \lambda^2 - x1 - x2$$
$$y3 = \lambda\ (x1 - x3) - y1,$$

where λ is equal to:

$$\lambda=(y2-y1)/(x2-x1) \text{ if } P \neq Q$$

and also:

$$\lambda=3x1^2+a/(2y1) \text{ if } P=Q$$

if $P = Q$; that means, the coordinates for 2P. Now there comes the moment to go on from the abstract thing – the points – to the concrete thing – the coordinates – of agreement with the previous definition of sum that evidently, it does not have anything to see with the supreme operation that the reader intuitively knows. It is the elliptical curve:

$$y^2=x^3+x+1$$

defined over $Z\_23=\{0,1,2,...22\}$. The points placed on this curve, will also belong to Z_23 and they are part of a finite set containing 27 points indicated in Table 2 and there exists a finite number of these points due to our choice. This fact is very important as computers are not able to work with infinite sets. Let P and Q be two points of our curve:

$$P=(3,10)$$
$$Q=(9,7)$$

Table 2. *Points over $y^2=x^3+x+1$ defined on the finite group Z_23*

-0,1	-0,22	-1,7	-1,16
-3,1	-3,13	-4	-5,4
-5,19	-6,4	-6,19	-7,11
-7,12	-9,7	-9,16	-11,3
-11,2	-12,4	-12,19	-13,7
-13,16	-17,3	-17,2	-18,3
-18,2	-19,5	-19,18	

Applying the previous formula, we see that `P+Q=(x _ 3,y _ 3)` is determined by doing the mathematical operations described here:

```
λ=11 which belongs to Z_23
x3=-6 = 17 (mod 23)
y3=89 = 20 (mod 23)
```

then P+Q gives, as a result, the coordinate point (17,20) placed on such a curve. Once the main characteristics for finite groups underlying elliptic curves have been analysed, it is possible to start with the concepts used for PKC based on such abstract entities. For ECC systems, there are three different approaches:

- Schema ECC Diffie-Hellman, based on Diffie-Hellman (1976) algorithm.
- Schema ECDSA, based on DSA or Digital Signature Algorithm.
- Schema ECMQC based on MQV approach.

Most importantly, it is the reason for the one that is possible to use these abstract entities in public-key cryptography that the security level of the ECDSA algorithms is the intrinsic difficulty of solving the discrete logarithm problem for elliptic curves (ECDLP) by means of brute-force attacks, which consists of the determination, given a curve E over `Z _ p`, the integer k between 0 `≤ l ≤ n -1` such as `Q=kP`, whenever this number does exist.

In order to avoid the proliferation of advanced topics and the subsequent overload for readers, we are going to focus now on ECDSA schemas and their application for cryptographic techniques in BSD OS, with a special emphasis on cipher and digital signature applications.

ECDSA Algorithm: A brief Introduction

ECDSA proposes a slightly analogous algorithm to DSA in which, rather than working with the set `Zp ={0,1,2,...p-1}`, the points over an elliptic curve with parameters `a, b ∈ Z_p` will be used instead.

Depending on the final use, ECDSA algorithm can be divided into two classes; one for cipher and the other one for signature generation. We discuss each of these classes separately.

Public/Private Keys Generation for ECDSA

The process of key pairs generation for ECDSA is based on the procedure detailed below:

- Choose an elliptic curve defined over a set of integers mod p, `Z _ p`. The number of points for the curve shall be divisible by a positive prime integer n.
- Select a point P over the elliptic curve whose order is n; that means, a point such as `nP=0`.
- Select an integer d within the interval `{1, 2, ..., n-1}`
- Calculate the point `Q=dP`

As a result, we get a pair of keys in which:

- Public Key is the set of values (E, P, n, Q)
- Private Key is the integer d, which belongs to the set `{1, 2, ... n-1}`

Let us study now the three schemas for cryptography we introduced before and their application to the basic problems in cryptography. These schemas can be extended to ECC algorithms and, in particular, to ECDSA algorithms; such as cipher of messages, and digital signature generation and verification.

Digital Signature Generation and Verification (ECDSA)

The proposal for ECDSA to get the digital signature for a message is summarised in the 6-step procedure below:

- Choose an integer number k belonging to the set `{1,2, ... n-1}`
- Calculate `kP=(x _ 1,y _ 1)`, and then set `r = x mod n`.
- If `r=0`, go to step 1, because if `r=0`, the equation `s=k^{-1}[h(m)+dr] mod n` will not contain the private key `d` and `a` valid d is required.
- Calculate `k^{-1} mod n`.
- Calculate `s=k^{-1}[h(m)+dr] mod n` where `h` is the resulting value provided by SHA-1 algorithm.
- If `s=0` go to step 1, to ensure the existence of an inverse `s^{-1}`.

Eventually, a digital signature, according to ECDSA, is nothing more than the pair of positive integers (r,s).

To verify a signature provided by ECDSA, the process consists of four sequential steps:

- Get a copy for public key's signature (E, P, n, Q) and verify that the signature (r,s) is included in the set of integers $\{1, \ldots, n-1\}$
- Calculate $w=s^{-1} \bmod n$ and $h(m)$.
- Calculate $u_1=h(m)w \bmod n$ and $u_2=rw \bmod n$
- Calculate $u_1P + u_2Q=(x_0,y_0)$ and $v=x_0 \bmod n$.

In this way, a digital signature will be valid whenever $v=r$ holds. Thus, we have a simple description of the possibilities of ECC to act as a replacement for traditional PKC algorithms.

Practical Applications, Small Privacy Guard (SPG)

Released in April 2009, version 0.3.1 of Small Privacy Guard appears as an application based on ECC cryptography, that can be used not only for the coding and deciphering of messages but also for digital signatures. SPG is the alternative to *Pretty Good Privacy* (PGP) of P. Zimmermann, developed from 1991, and the Open Source versions like OpenPGP and GnuPG.

This application is freely available at the URL *http://spg.sourceforge.net* in the form of source code under the terms of the Lesser General Public License version 2.0 (LGPLv2) for download. Unlike PGP, the keys are not linked to email addresses or user's names.

The power of this application, as any product that admits ECC cryptography, is based on the number and complexity of the implemented elliptic curves. In the case described in this article, SPG has 11 such curves:

- secp112r1
- secp112r2
- secp128r1
- secp128r2
- secp160r1
- secp160r2
- secp192r1
- secp224r1
- secp256r1
- secp384r1
- secp521r1

whose key sizes vary from 112 to 521 bytes.

The reader acquainted with GPG will not have a problem in adapting SPG as an alternative tool in cryptographic applications in Unix-like systems such as NetBSD. To start with the installation process, the following requirements must be satisfied:

- OpenSSL Development libraries.
- gcrypt development libraries.

For NetBSD distributions, the installation process is automated thanks to pkgsrc system, thus there is no need to install utilities such as OpenSSL and Qcrypt. Otherwise, these libraries should be installed before downloading the source code for SPG application. If you want

Listing 1. *Standard test cases provided by SPG application*

```
$ make tests
...
INFO: File encrypted
Message encrypted ok
######### secp521r1 decrypting data #################
INFO: File decrypted successfully
Message decrypted ok
INFO: File decrypted successfully
Message decrypted ok
ALL TESTS PASSED
```

Listing 2. *Test cases execution for SPG application*

```
gcc -g -Wall -O2 -c curves.c
gcc -g -Wall -O2 -c ecc.c
gcc -g -Wall -O2 -c ec_point.c
gcc -g -Wall -O2 -c main.c
gcc -g -Wall -O2 -c utils.c
gcc -g -Wall -O2 -c sym_cipher.c
gcc -g -Wall -O2 -c help.c
gcc -o spg curves.o ecc.o ec_point.o main.o utils.o
sym_cipher.o help.o -lgcrypt -pthread -lssl -lcrypto
echo “done”

cp spg tests/ && cd tests && ./tests.sh
```

Listing 3. *SPG application usage and options*

```
Use: spg commands [options] [file ...]
Commands are:
-g --gen_key Generate private key
-x --xport eXport public key from private key
-s --sign Generate message signature
-v --verify Verify message signature
-e --encrypt Encrypt
-d --decrypt Decrypt
-l --list_curves List implemented curves
-h --help Print help and exit
Options are:
-c --curve Use this curve
-i --input Input file
-o --output Output file
-V --verbose Be loud
-t --timing Print time spent computing ecc algorithms
```

to perform an installation from the source, uncompress the downloaded file by issuing the commands and starting the build process:

```
$ tar zxvf spg.tar.gz && cd spg
$ make all
```

It is recommended to execute the tests in order to check if the application works properly (see Listing 1). To test SPG, execute the command and verify that all proposed tests are successfully passed. Once this step is finished, it is time to get started with the first steps using ECC cryptography by means of SPG to examine its use and features for key generation, crypt and decrypt, as well as digital signature management, which were introduced for ECDSA algorithm (Listing 2). The SPG application uses a command-line interface (CLI) that allows the commands and options shown in Listing 3. Eventually, we are going to analyse SPG within the three basic processes for Digital Cryptography.

ECC Key Generation for SPG

All processes based on ECC shall use one of the 11 elliptic curves currently implemented in SPG 0.3.1. To generate a pair of keys using, for instance, the curve `secp521r1` which provides the strongest security level, the command is:

```
$ spg -g -c secp521r1 -o ecc.key
```

This command stores the private key in an ASCII file `ecc.key` (see Listing 4). Once our private key has been generated, its complementary public key shall be obtained by means of the command:

```
$ spg -x -k ecc.key -o ecc_pub.key
```

which stores it in the file `ecc _ pub.key`, whose contents are reproduced below. Notice that the elliptic curve does not need to be specified again once the private key has been generated (Listing 5).

In this way, we have already the necessary elements to develop the most common applications for cryptography. The only thing left to do is to assign each pair of keys to a specific user.

Crypt/Decrypt with SPG

Using the public key we created in the previous section to encrypt a single file, let's say `myfile.c`, we can cipher the file into one with the extension .enc as shown below:

```
$ spg -e -k ecc_pub.key myfile.c
INFO: File encrypted
```

Before decrypting the file, rename it in order to delete the .enc extension:

```
$ file myfile.c*
myfile.c: ASCII English text
myfile.c.enc: data
```

The decryption process is quite similar, just swap the public and private ECC keys:

```
$ spg -d -k ecc.key -o myfile.c myfile.c.enc
INFO: File decrypted successfully
```

The procedures concerning SPG installation and basic operations can be applied to a wide variety of Unix systems and not only for our target BSD OS in a very straightforward way.

Listing 4. *ECC Private Key based on elliptical curve secp521r1*

```
-----BEGIN SPG PRIVATE KEY-----
QQYIehs0rMVKFNPcgiXFHJT+H6lOupm54ICJhAxkUja5vzeJPZBpqkdHeWSke5oE
Xi0KswHyU8lg14Ap7R08BOESQgHU+wr0Q4dfvltuFzSjTxj15cVeFhfq4xhrVxxn
86LsISGw8oeTeAab3scOiSxdjTD3hqHWdRfVwnjN2+tOk6Ez40F/rpqFeU5iTgMV
rNCB8IJmcOxB0J9nwWlWuyD3rBVrP8ox6tSIHEKaiVR+0ogyJL3oQ78LfIW8hl4d
QLQaJtOD5QlzZWNwNTIxcjE=
-----END SPG PRIVATE KEY-----
```

Listing 5. *ECC Public Key based on elliptical curve secp521r1*

```
-----BEGIN SPG PUBLIC KEY-----
QQYIehs0rMVKFNPcgiXFHJT+H6lOupm54ICJhAxkUja5vzeJPZBpqkdHeWSke5oE
Xi0KswHyU8lg14Ap7R08BOESQgHU+wr0Q4dfvltuFzSjTxj15cVeFhfq4xhrVxxn
86LsISGw8oeTeAab3scOiSxdjTD3hqHWdRfVwnjN2+tOk6Ez4wlzZWNwNTIxcjE=
-----END SPG PUBLIC KEY-----
```

Digital Signature and Verification (ECC) with SPG

To conclude with the ECC applications provided by SPG application, we describe the use of digital signatures for files and the subsequent verification. Let us take as an example the file `message.txt`, whose contents may be found in Listing 6.

By using our private key, the process to get a digital signature associated to this file, is given by the command:

```
$ spg -s -k ecc.key -o signature message.txt
```

The resulting ECC-based digital signature is shown in Listing 7.

Obviously, to check that the file has not been altered, you have to use the public key provided by SPG by typing the command:

```
$ spg -v -k ecc_pub.key -i signature message.txt
INFO: Signature is valid
```

Hence we finish the study of application examples for SPG to illustrate the application of ECDS algorithms in Unix-like systems.

Another useful application: OpenSSH and ECC Usage

Although there is no reason to fear possible weaknesses with RSA/DSA methods, the ECC may provide an additional method to generate public/private keys. Likewise, ECC and the ECDSA support, it is a different system to avoid the existence of a single point of failure in case a security breach occurs.

As NetBSD 6.0 incorporates OpenSSH 5.9 together with ECSDA support and OpenSSL 1.0.1c by default, the generation of a pair of user's keys follows the same procedure used for RSA/DSA keys with no need for additional recompilation for OpenSSH utilities (Listing 8).

The key's randomart image is shown on Listing 9.

As you can see, the use of ECC methods for SSH authentication is really straightforward and there is no additional effort to generate alternative authentication based on elliptic curve cryptographical methods.

ECC and Current Standards

It is required to digress a bit and introduce some concepts on the standardization of these algorithms, in order to promote the interoperability and facilitate the use of technologies currently in use. Standards for ECC are currently being prepared by some international organizations:

ECC algorithms are being standardised by the American National Standards Institute (ANSI), more precisely by the ASC X9 (Financial Services) working group, and there are two documents: ANSI X9.62, The Elliptic Curve Digital Signature Algorithm (ECDSA) and ANSI X9.63, Elliptic Curve Key Agreement and Transport Pro-

Listing 6. *Contents of message.txt file*

```
$ cat message.txt
Recuerde el alma dormida,
avive el seso y recuerde
contemplando:

Cómo se pasa la vida,
cómo se viene la muerte
tan callando.
```

Listing 7. *Digital signature for message.txt*

```
$ cat signature
-----BEGIN SPG SIGNATURE-----
QdpgUmdMn+5IFH+1MCVx9WcjrHT8fmKdu4LB4jn/UxmYa2if4vfSWugq7W59tv3J
texOuBez6mj7AXiu58QKLcYBQZ9FQrfrkVZ4tESoQ0NujxQYkXKeBesyb2m9Ydir
IdcmNXyWFQ4RRZ4VeuAc7siaJDUT/fJT10+aZkLOmpx133Ze
-----END SPG SIGNATURE-----
```

Listing 8. *Generating a key*

```
# ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ecdsa.
Your public key has been saved in /root/.ssh/id_ecdsa.pub.
The key fingerprint is:
72:4d:60:c4:8f:3a:4d:da:bb:00:7c:e6:7f:fb:ec:7f root@dhcppc0
```

Listing 9. *The key's randomart image*

```
+--[ECDSA 256]---+
| o+ |
| ... |
| o. |
| . oo. |
| o +*S . |
| =+oo |
| o. . |
| o. .. E|
| oo.++... |
+-----------------+
```

tocols. Additionally, elliptic curves have also been introduced into the specifications for PKC algorithms.

On the other hand, the Internet Engineering Task Force (IETF) describes a modified protocol to accept/reject keys which is a variant of Diffie-Hellmann algorithm using ECC.

The International Standards Organisation (ISO) is working in Appendix 3 included in ISO/IEC 14888: Digital signature with appendix, which is mainly devoted to the certificate-based mechanisms. These mechanisms also support ECC.

The *Asynchronous Transfer Mode* (ATM) Forum Technical Committee's Phase I ATM Security Specification draft document tries to introduce new mechanisms to reinforce ATM network security, by providing the necessary support for a wide variety of public-key algorithms such as RSA, DSA, and indeed ECDSA.

To sum up, it is clear that the family of ECC algorithms are serious candidates to replace the traditional PKC algorithms used in the industrial, financial, and science sectors in cases where a better performance at the same security level is required.

Conclusion

As indicated in the introduction, the technologies of digital cryptography, in general and especially those related to Public Key methods, are based on a family of mathematical problems represented by abstract elements and entities that only can be clearly understood by using higher mathematics.

Concretely, the main advantage of the ECC algorithms arises from the low resource requirements at the moment of doing the coding, deciphering, signatures and checking operations of these, as well as the total independence of the platform and most importantly, the public availability of the algorithms, which are in the phase of adoption by the most important international agencies of standardization.

The ECC algorithms provide, as explained in this article, among all existing public-key algorithms, the highest degree of protection with the smallest key size in comparison with the public key algorithms used at present. With a module of 160 bits, an algorithm ECC as ECDSA offers the same level of security as a DSA/RSA with 1024 bit-module. Working with keys of low size saves bandwidth, involves smaller digital certificates, and leads to quicker algorithm implementations with lower resource consumption and hardware requirements.

For those readers interested in a more in-depth study of elliptic curves, I recommend the latest book by Avner Ash and Robert Gross, "Elliptic Tales: Curves, Counting and Number Theory", which describes the latest developments on this topic and provides detailed explanations about the underlying theory of elliptic curves.

References

- National Institute for Standards and Technology, "Digital signature standard", FIPS Publication 186, 1993. Available at *http://csrc.ncsl.nist.gov/fips/*
- A. Odlyzko, "The future of integer factorization", CryptoBytes – The technical newsletter of RSA Laboratories, volume 1, number 2, Summer 1995, 5-12. Available at http://www.rsa.com/
- B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd edition, Wiley, 1996.
- D. Stinson, Cryptography = Theory and Practice, CRC Press, 1995.

Acronyms and Abbreviations

ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
DSA	Digital Signature Algorithms
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm
ECDS	Elliptic Curve Digital Signature
HA	Secure Hash Algorithms
IETF	Internet Engineering Task Force
ISO	International Standards Organisation
MD	Message Digest
PGP	Pretty Good Privacy
PKC	Public Key Algorithms
RPM	RedHat Package Manager
RSA	Rivest-Shamir-Adleman Algorithm
SKC	Symmetric Key Algorithms

JOSÉ B. ALÓS

José B. Alós began his professional career in 1999 with EDS, as a UNIX System Administrator mainly focused on SunOS/Solaris, BSD and GNU/Linux. Five years ago, he joined EADS Defense and Security. Nowadays he works for CASSIDIAN where he is responsible for providing end-user support in aircraft engineering departments for long-term projects. That is the main reason behind this article as VAX/VMS systems play a paramount role in today's aerospace industry for a wide variety of embedded RT systems conceived for mission and flight operations. He was also an Assistant Professor in the Universidad de Zaragoza (Spain), specializing in the design of High Availability solutions and his academic background includes a PhD in Nuclear Engineering and three MsC degrees in Electrical and Mechanical Engineering, Theoretical Physics, and Applied Mathematics.

SPRING FUNDRAISING DRIVE

Support FreeBSD by donating

Your donations have helped make FreeBSD the best OS available! By investing in the services provided by The FreeBSD Foundation you have helped us fund projects to keep FreeBSD a high-performance, secure, and stable OS.

What will the Foundation accomplish with your donation in 2013?

- **Software development projects** for FreeBSD: **$600,000.**
- Paid staff time supporting **Release Engineering and Security teams.**
- **Grow staff: Five technical staff members** by year-end.
- **Provide support for BSD conferences around the globe,** in Europe, Japan, Canada, and the USA.
- **Hardware** to maintain and improve FreeBSD project infrastructure: **$130,000.**
- **FreeBSD community growth** through marketing and outreach to users and businesses.
- **Legal services and counsel protecting the FreeBSD trademarks.**

FreeBSD is internationally recognized as an innovative leader in providing a high-performance, secure, and stable operating system. Our mission is to continue and increase our support and funding to keep FreeBSD at the forefront of operating system technology. But, we can't do this without your help!

Last year with your generosity, we raised over $770,000. This year we will invest $1,000,000 to support and promote FreeBSD.

We have kicked off the new year with three newly funded projects, and are actively soliciting additional project proposals.

Please support the Foundation during our Spring Fundraising Drive, and help us raise $100,000 from 1000 donors between April 15th and May 30th.

We need your help.

We can't do this without you...

Make your donation today. Go to:
www.freebsdfoundation.org/donate

The FreeBSD FOUNDATION

Then talk to your employer about matching your gift— or making their own donation.

Find out more, visit:

www.freebsdfoundation.org

iXsystems®

Headquarters:
San Jose, CA

855.GREP.4.IX | Contact Us

99% Compatibility

online now...

IXSYSTEMS AND YOU ARE THE PERFECT MATCH

SHARED INTERESTS

- ☑ Enterprise Storage Solutions
- ☑ Personalized Customer Service
- ☑ Bold New Information Technology

I'm a Storage Reseller

In The EU

Looking for Storage Solutions to Sell

A Technology Partner
More Technical Experience
New Business Opportunities

Visit Today!

iXsystems

Technology Partner Seeking Resellers/Integrators for TrueNAS™ Storage Appliance

WWW.IXSYSTEMS.COM/PERFECTMATCH